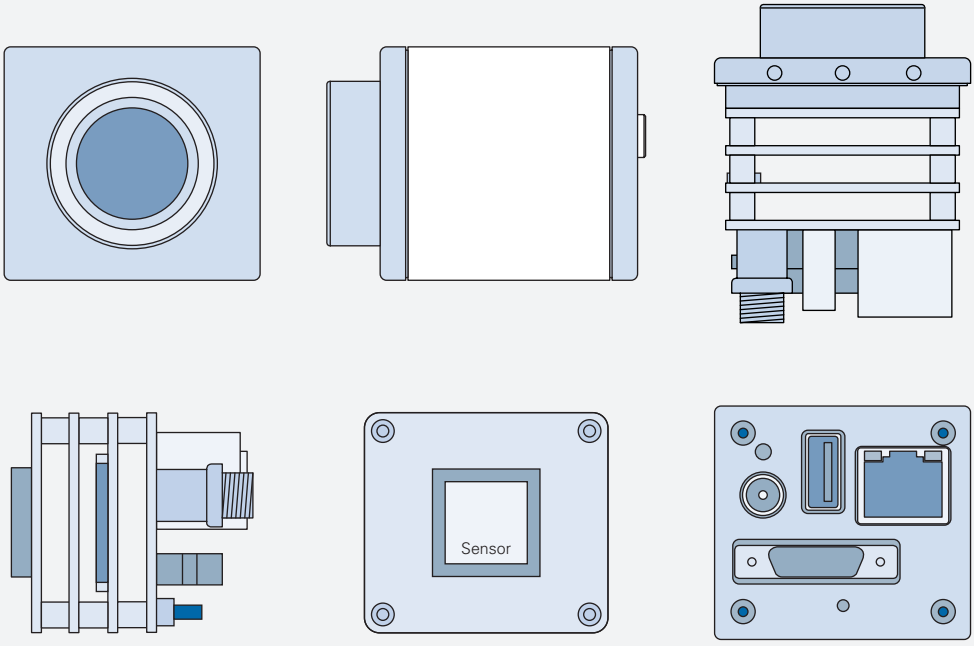
VRmagic
D2 Intelligent Camera
User Guide



Issued August 2014

VRmagic

VRmagic GmbH

Augustaanlage 32
68165 Mannheim
Germany

Phone +49 (0)621 400 416-20
Fax +49 (0)621 400 416-99

info.imaging@vrmagic.com
www.vrmagic-imaging.com

D2 Intelligent Cameras – User Guide

Document version: 1.15
Date of issue: August 14, 2014
Applicable to: all D2 intelligent cameras, SDK release 4.1

Subject to change without notice. Errors excepted.

This document is protected by copyright. All rights reserved. No part of this document may be reproduced or transmitted for any purpose in any form or by any means, electronically or mechanically, without expressly written permission by VRmagic.

Windows® is a registered trademark of Microsoft®.
DaVinci™ is a trademark of Texas Instruments.

Table of Contents

1 General Information 6

2 Safety 7

3 Overview 8

3.1 D2 Intelligent Camera Architecture 8

3.2 Camera Models 10

3.3 Connectors and Interfaces 13

3.4 Accessories 15

4 Software Installation 16

4.1 Downloading the VRmagic Software 16

4.2 Installing the D2 Camera SDK 16

4.3 Installing the Camera Runtime and the Host SDK 18

4.4 Installing the USB to RS232 Driver 19

4.5 Installing a Console Application 20

5 Hardware Installation 21

5.1 Connecting External Sensor Boards 21

5.2 Connecting an External HDMI Board 23

5.3 Connecting the Interface Cables 25

5.3.1 Standard Interface Board 25

5.3.2 CUEO1 Pico Interface Board 27

5.3.3 Connecting Serial Console and JTAG 28

5.4 Connecting the Power Supply 29

6 First Steps 30

6.1 Setting Up the Ethernet Connection 30
6.1.1 Setting Up a Point-to-Point Connection 31
6.1.2 Enabling Ethernet Support and Checking the Connection 31
6.2 Accessing the Device via SSH 32
6.3 Accessing the Device via RS232 34
6.4 Changing the Camera's Default Password 35
6.5 Changing the Camera's IP Address 35
6.6 Exchanging Files and Data 36
6.7 Changing DSP Memory Size 38
6.8 Changing Kernel Boot Parameters 39
6.8.1 Partitioning Linux and Shared Memory (CMEM) 40
6.8.2 Changing Video Output Mode 41
6.9 Camserver and CamLab 42
6.10 Accessing the Camera's GPIOs 43
6.11 Accessing the Camera's Temperature Sensor 43
6.12 Using Trigger and Strobe 44
6.12.1 Trigger Input 44
6.12.2 Strobe Output 45
6.13 Switching the Camera On and Off 46

7 Programming Environment 47

7.1 Setting Up the ARM Toolchain 48
7.2 Setting Up the DSP Toolchain 50
7.3 Update of an Older Development Kit 54
7.4 Running Demo Applications 54
7.4.1 ARM Demos 54
7.4.2 ARM and DSP Demos 56

8 Firmware Update and Recovery 60
8.1 General Description 60
8.2 Updating via Ethernet 61
8.3 Updating or Recovering via USB Update Flash Drive 64
8.3.1 Generating a USB Update/Recovery Flash Drive 65
8.3.2 Manual Update 67
8.3.3 Automatic Update 68

9 Appendix 70
9.1 Cable Plans and Pinouts 70
9.1.1 VRmDC-X Interface Adapter Cable 70
9.1.2 VRmDC-2 Service Cable 72
9.1.3 VRmCUEO1 Pico Interface Board 73
9.2 Linux Short Reference and Examples 76

10 Index 77

1 General Information

This guide applies to all intelligent cameras based on the D2 intelligent camera platform from VRmagic. Follow this guide chapter by chapter to set up and understand your device. If a section of this document only applies to certain camera models, this is indicated at the beginning of the respective section.

Symbols Used

This guide makes use of the following symbols and conventions:

Warning!

Indicates a situation which, if not avoided, could result in minor or moderate injury and/or property damage or damage to the device.

Caution

Indicates a situation which, if not avoided, may result in minor damage to the device, in malfunction of the device or in data loss.

Note

Notes provide information on special issues related to the device or provide information that will make operation of the device easier.

This headline indicates the beginning of a procedure:

1. This number indicates the first step of a certain procedure you are expected to follow. Following steps are numbered accordingly.
   - ▶ This arrow indicates an expected result of your action.
     `Linux commands are written in this monospaced font.`

➲ This symbol indicates a reference to a different part of this manual or to external documents.

2 Safety

The declarations of conformity (as far as available or applicable), data sheets, and operating instructions applicable to your product are an integral part of this document. Keep these instructions for future use.

Intended Use

The device comprises an embedded PC with integrated camera. The device is designed for capturing, processing, and transmission of (image) data and analog or digital signals.

General Safety Instructions

Installation and operation may only be carried out by trained personnel.

Faultless and safe operation of the device requires proper transport, storage, setup, and handling.

Please refer the device to VRmagic if repairs should become necessary. Servicing is required if the device has been damaged in any way or does not operate normally.

The camera generates heat during operation. Make sure that the camera is always operated within the specified temperature range. If necessary, provide adequate heat dissipation measures.

Do not subject the device to direct sunlight, wet conditions, dust, strong impacts, and heat. Observe the ambient conditions specified in the following.

Unplug the device during lightning storms or when not in use for longer periods of time.

Do not disassemble any part of the camera, unless explicitly described in this manual. Disassembly may cause damage and voids the product warranty.

Power Supply

Connect the device to an appropriate power supply only (5 V DC ±5%, 15 W).

Ambient Conditions

Do not operate the device near a TV or radio station that radiates high powered radio-frequencies. Operating temperature: 0...40°C (32...104°F). Storage temperature: -30...80°C (-22...176°F).

ESD Prevention

Electrostatic discharges may cause serious, permanent damage to the camera. When working with board-level cameras, always make sure that you work in an ESD protected area according to international standards (IEC 61340-5-1 (2007-08)).

3 Overview

3.1 D2 Intelligent Camera Architecture

The D2 intelligent cameras from VRmagic function as an embedded system. They are equipped with a highly integrated DaVinci™ Dual-Core CPU from Texas Instruments, which features a 300 MHz ARM9 processor and a 600 MHz digital signal processor (DSP).

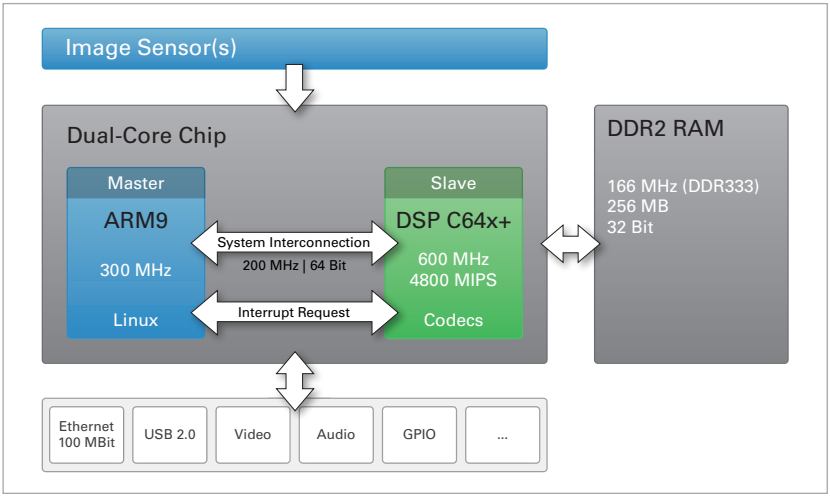


Fig. 1: Intelligent camera architecture (D2 platform)

The Linux OS runs on the ARM processor so that the DSP is completely available for image processing tasks. Communication betwen ARM and DSP is controlled by the DSPlink kernel module.

We recommend the Codec Engine framework from TI to the application developers who wish to use both the ARM and the DSP for their applications.

By default, the camera's DDR2 memory is partitioned into three areas: the Linux memory, the contiguous shared memory (CMEM), and the DSP memory. Images from the image sensor are always written to the shared memory, where they are accessible for both the ARM and the DSP.

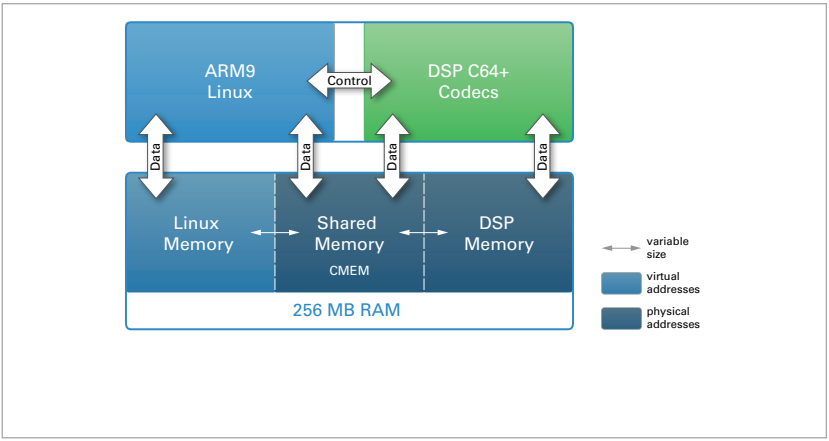


Fig. 2: Interaction of dual-core CPU and memory

The default size of the memory partitions are 96 MB for the Linux memory, 96 MB for the shared memory, and 64 MB for the DSP memory. The size of the memory areas can be changed if necessary (➲ section 6.7 and section 6.8).

3.2 Camera Models

Single-Sensor Cameras VRmD(F)C-X

Single-sensor cameras are equipped with one image sensor. The sensor board is mounted directly onto the camera base unit, thus forming one compact unit. Single-sensor cameras are available as board-level cameras (OEM versions), board-level cameras with optics (COB versions) and as cameras with aluminium housing (PRO versions).

All cameras are equipped with a sensor board, a CPU board, and an interface board. Depending on the image sensor, the camera may be equipped with an additional FPGA board (VRmDFC-X models).

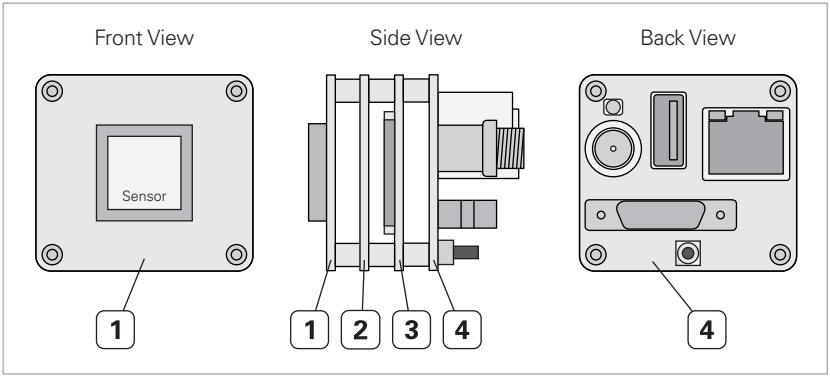


Fig. 3: Intelligent single-sensor camera (OEM) with standard interface board and FPGA board

1 Sensor board with image sensor
2 FPGA board (VRmDFC-X cameras only)
3 CPU board
4 Interface board

Remote-Sensor Cameras VRmDC-X-E

Remote-sensor cameras consist of a base unit and an external sensor board connected to the base unit by flex-foil cable. Remote-sensor cameras are available as board-level cameras (OEM versions) and board-level cameras with optics (COB versions).

The base unit consists of a sensor connection board, a CPU board, and an interface board.

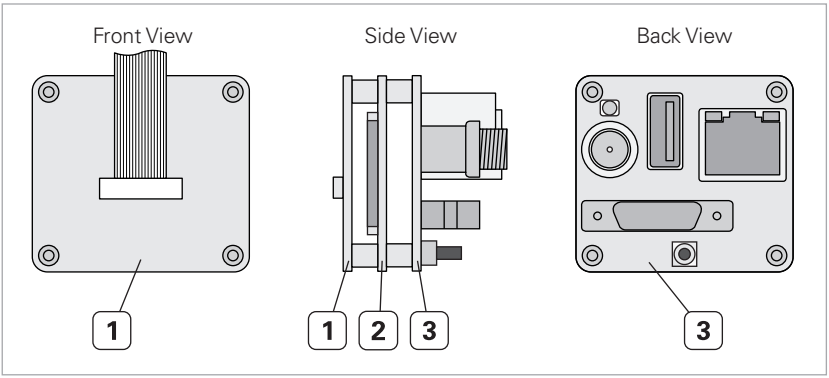


Fig. 4: Base unit of intelligent remote-sensor camera with standard interface board

1 Sensor connection board (external sensor board is connected via flex-foil cable)

2 CPU board

3 Interface board

Multi-Sensor Cameras VRmDMFC

Multi-sensor cameras consist of a base unit and up to four external sensor boards connected to the base unit by flex-foil cables. They are available as board-level cameras (OEM versions) and board-level cameras with optics (COB versions).

The base unit consists of a sensor connection board, an FPGA board, a CPU board, and an interface board.

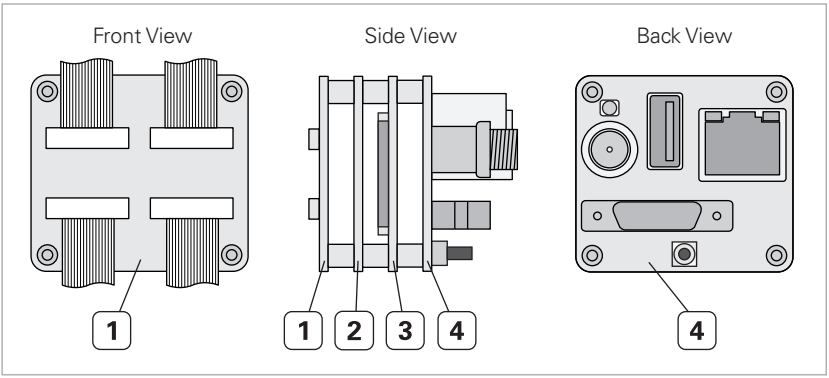


Fig. 5: Base unit of intelligent multi-sensor camera with standard interface board

1 Sensor connection board (external sensor boards are connected via flex-foil cables)

2 FPGA board

3 CPU board

4 Interface board

3.3 Connectors and Interfaces

All board-level cameras (OEM and COB versions) are available with two different interface boards: the standard interface board and the CUEO1 pico interface board. PRO cameras in aluminium housing are available only with the standard interface board.

- Standard interface board: This board is equipped with standard connectors, for example an RJ45 receptacle for the Ethernet connection.
- CUEO1 pico interface board: All interfaces are available on miniature connectors (Molex PicoBlade), thus reducing board size and weight. This board allows for cost-effective and flexible connections.

All board-level cameras (OEM and COB versions) feature a lateral DF14-15P connector for serial console and JTAG located on the CPU board. This connector is not accessible in housing cameras (PRO versions).

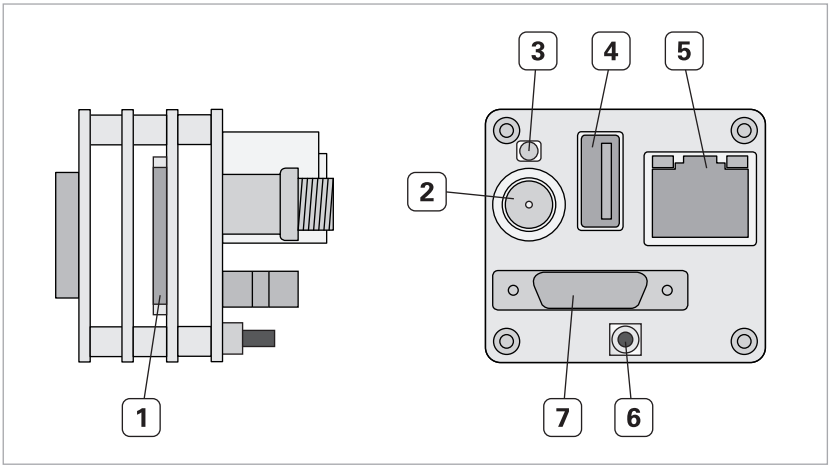


Fig. 6: OEM camera with standard interface board

1 DF14-15P connector for serial console and JTAG on CPU board
2 Power inlet (+5 V DC, ±5%)
3 Red status LED
4 USB host connector
5 RJ45 Ethernet connector
6 Power button
7 MPE Garry Micro T connector for trigger, strobe, RS232, S-Video, GPIOs

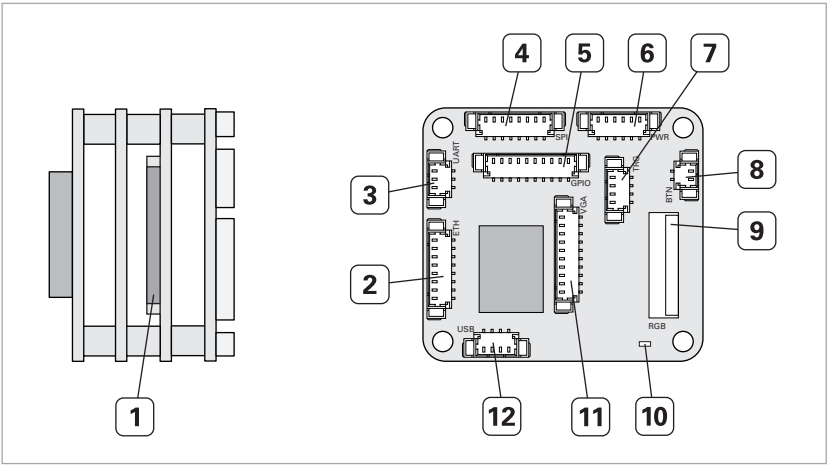


Fig. 7: OEM camera with CUEO1 pico interface board

1 DF14-15P connector for serial console and JTAG on CPU board
2 Ethernet
3 Serial port (UART)
4 SPI port
5 General porpose I/Os
6 Power inlet (+5 V DC, ±5%)
7 Trigger and strobe
8 Power button
9 RGB888 interface for connection of external HDMI output board
10 Green status LED
11 Analog video output (Composite or S-Video)
12 USB host connector

3.4 Accessories

There are optional accessory kits for the D2 intelligent cameras containing adapters, cables, power supply and a USB update flash drive.

VRmDC Accessory Kit 2.0 For cameras with standard interface board	
1 VRmDC-2 service cable	1 USB-serial adapter 9-pin
1 VRmDC-X adapter cable	1 USB update flash drive
1 AC Adapter 5 V/15 W Worldwide	1 S-Video cable
1 Null modem cable 3 m	

VRmDC Accessory Kit PICO For cameras with pico interface board	**VRmDC Accessory Kit PICO2** For cameras with pico interface board
1 VRmDC-2 service cable	1 VRmDC-2 service cable
1 AC Adapter 5 V/15 W Worldwide	1 AC Adapter 5 V/15 W Worldwide
1 Null modem cable 3 m	1 Null modem cable 3 m
1 USB-serial adapter 9-pin	1 USB-serial adapter 9-pin
1 USB update flash drive	1 USB update flash drive
1 Picoblade-open end 2 pin	1 Picoblade-open end 2 pin
1 Picoblade-open end 3 pin	1 Picoblade-open end 3 pin
2 Picoblade-open end 4 pin	1 Picoblade-open end 4 pin
1 Picoblade-open end 6 pin	1 Picoblade-open end 8 pin
2 Picoblade-open end 8 pin	2 Picoblade-open end 10pin
2 Picoblade-open end 10pin	1 Ethernet Adapter Cable VRmCUEO1
	1 Ethernet Adapter Panel (RJ45)
	1 USB-A Adapter Cable VRmCUEO1
	1 Power Cable VRmCUEO1

4 Software Installation

4.1 Downloading the VRmagic Software

Downloading the software package:

1. Visit www.vrmagic.com/imaging/downloads/ .
2. Scroll down to **D2 Intelligent Camera Platform**.
3. Download the .zip file **VRmUsbCam Distribution for Developers** and keep it for future use.

4.2 Installing the D2 Camera SDK

VRmagic D2 Camera SDK

The D2 SDK is required to develop camera applications on your host PC that will be executed on the D2. To do so, you need to additionally set up and configure cross-compile tool chain on your host PC as described in ➲ chapter 7.

Installing the VRmagic D2 Camera SDK:

1. To install the SDK, unzip the file you downloaded in the previous section.
   ▶ The following directories were created on your hard drive.

Directory structure of the D2 SDK distribution (excerpt)	
D2	
development_kit	VRmUsbCam DevKit for D2
demos	Examples for the API usage
docs	Documentation of the VRmagic camera API
include	VRmagic API include files
lib	VRmagic API development library
toolchains	Manuals and software regarding ARM and DSP Cross-Compilers
vm_lib	VRmagic machine vision library
wrappers	C++ wrapper
docs	Additional documentation (cable plans and pinouts)
drivers	Windows driver for the USB-serial adapter, Windows driver for accessing ext2 partitions
firmware	Binary images of the camera's firmware (bootloader, kernel and rootfs)
src	
kernel	Linux kernel sources
rootfs	tar archive of the camera's root filesystem
linux	VRmUsbCam DevKit for Linux host PC (x86)
windows	VRmUsbCam DevKit for Windows host PC (x86)

4.3 Installing the Camera Runtime and the Host SDK

VRmagic Windows/Linux PC SDK

The host PC SDK is required to develop camera applications that will be executed on the host PC. Thus, this SDK is not required for compiling camera applications that will be executed directly on the D2. However, the host SDK is useful to debug your camera application on the host PC first and cross-compile it for the D2 later.

VRmagic Windows/Linux PC Camera Runtime and CamLab Application

The camera runtime enables the Linux host PC to access VRmagic cameras, for example using the VRmagic CamLab application. CamLab enables you to find, test and configure all VRmagic cameras available on the Ethernet.

Note

The included SDK, the runtime libraries for the host PC, and the cross-compile toolchain for ARM by Mentor Graphics are built for the x86/i386 architecture (32 bit).

In order to install them on a 64 bit machine, you have to make sure that your operating system supports the installation of 32 bit applications. Windows supports this by default. On 64 bit Linux machines the support for 32 bit applications needs to be activated by the user. Please consult the documentation of your Linux distribution for more information.

Installing the camera runtime and SDK for Windows:

1. Navigate to the **windows** directory in the downloaded and unzipped SDK distribution.
2. Unpack the zip file **VRmUsbCam DevKit for Windows (x86)*.zip**.
3. To install the SDK, run **setup.exe** from the directory you unzipped in step 2.
4. Follow the on-screen instructions.
   - A message is displayed after successful installation. A new folder **VRmagic** was created in your Windows start menu.

Installing the camera runtime and SDK for Linux:

1. Navigate to the **linux** directory in the downloaded and unzipped SDK distribution.
2. Navigate to the **x86** subdirectory.
3. To install the runtime libraries and the CamLab application, execute
   `sudo dpkg -i vrmagic-linux-pc-camera-runtime-*.deb` .
4. To install the host PC SDK, execute
   `sudo dpkg -i vrmagic-linux-pc-camera-sdk-*.deb` .
5. For further information read the README file in the directory **/opt/vrmagic/sdk-*/x86/development_kit**.

4.4 Installing the USB to RS232 Driver

Note

You will only need the USB to RS232 adapter driver if you have a board level camera (OEM or COB) and a Windows computer that does not have a serial port.

The driver for Linux is already included in the kernel main line (pl2303), so your Linux distribution should work with the USB to RS232 adapter out of the box.

Installing the USB to RS232 adapter driver:

1. Navigate to the **D2/drivers/usb_to_rs232_adapter** directory in the downloaded SDK distribution.
2. Unpack the file **USB_to_Serial_cable.zip**.
3. To install the driver, run the included .exe file and follow the on-screen instructions.

4.5 Installing a Console Application

We recommend installing PuTTY for Windows computers. PuTTY is an open-source SSH, Telnet, and serial client. The standard Windows console does not support SSH, which is necessary to access the camera.

Note

Linux users can use the Linux terminal to access the camera using SSH. However, we also recommend using PuTTY under Linux, because it is more convenient.

Downloading and running PuTTY (Windows):

1. Visit http://www.putty.org/ .
2. Download **putty.exe**.
3. No installation is necessary. Simply run **putty.exe** to use the console.

Downloading and running PuTTY (Ubuntu Linux):

1. Install PuTTY by executing `sudo apt-get install putty` in the terminal.
2. To run PuTTY, execute `sudo putty` .

5 Hardware Installation

5.1 Connecting External Sensor Boards

Note

This section only applies to remote-sensor cameras and multi-sensor cameras.

Caution
Possible malfunction of camera

Observe the correct orientation of the flex-foil cable contacts as indicated in the instructions and the drawings. Otherwise the device will not function.

Connecting the flex-foil cable(s) to the base unit:

1. Pull out the lock of the connector on the base unit (➲ Fig. 8 on page 21).
2. Insert the flex-foil cable with the blue marker facing the board and the open contacts facing you.
3. Secure the cable by pushing the lock back in.
4. Repeat steps 1 to 3 for further external sensor boards.

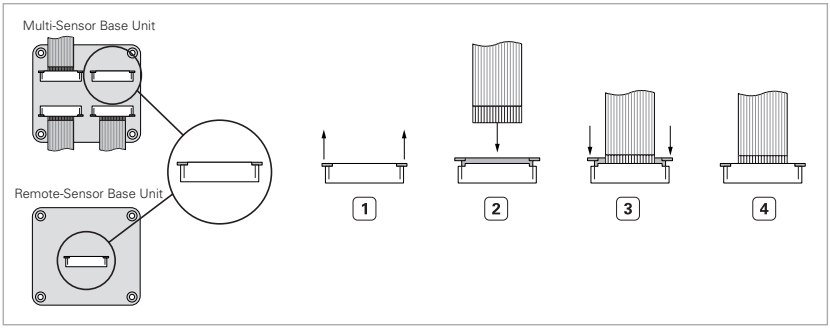


Fig. 8: Connecting the flex-foil cable to the base unit: contacts visible, blue marker not visible

Connecting the flex-foil cable(s) to the sensor board(s):

1. Pull out the lock of the connector on the sensor board (➲ Fig. 9 on page 22).
2. Insert the flex-foil cable with the blue marker facing you and the open contacts facing towards the sensor board.
3. Secure the cable by pushing the lock back in.
4. Repeat steps 1 to 3 for further external sensor boards.

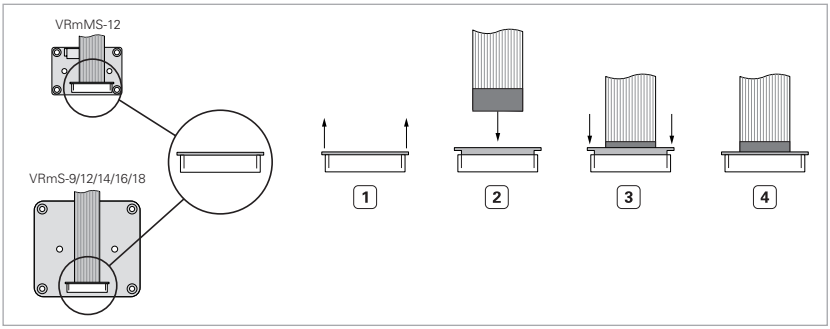


Fig. 9: Connecting the flex-foil cable to the sensor board: contacts not visible, blue marker visible

5.2 Connecting an External HDMI Board

Note

This section only applies to cameras equipped with the pico interface board VRmCUEO1 and an optional external HDMI output board.

Warning!
Possible destruction of equipment

The camera stack and the HDMI board will be damaged should the FPC cable be inserted with the wrong orientation while the camera is connected to power!

Observe the correct orientation of the FPC cable contacts. The visible electrical contacts and the label of the cable have to face the boards (➲ Fig. 10 on page 23).

The VRmCUEO1 pico interface board is equipped with an RGB888 interface via a 45-pin flexible printed circuit connector (➲ Fig. 7 on page 14). Use this interface to attach the VRmagic HDMI output board to the intelligent camera. An FPC cable is supplied.

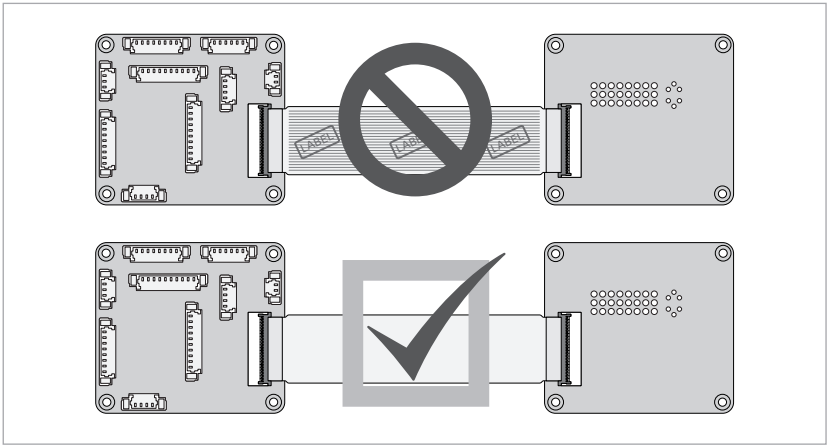


Fig. 10: Correct orientation of the flex-foil cable

Connecting an external HDMI board:

1. Open the lock of the FPC connector on the camera interface board by flipping it upwards (➲ Fig. 11 on page 24).
2. Insert the FPC cable observing the correct orientation. The visible electrical contacts and the label have to face the board (➲ Fig. 10 on page 23).
3. Secure the cable by pushing the lock down.
4. Proceed the same way to connect the cable to the HDMI board.

Note

The boot parameters of the camera have to be changed in order to make use of the HDMI board. A detailed description is provided in ➲ section 6.8.

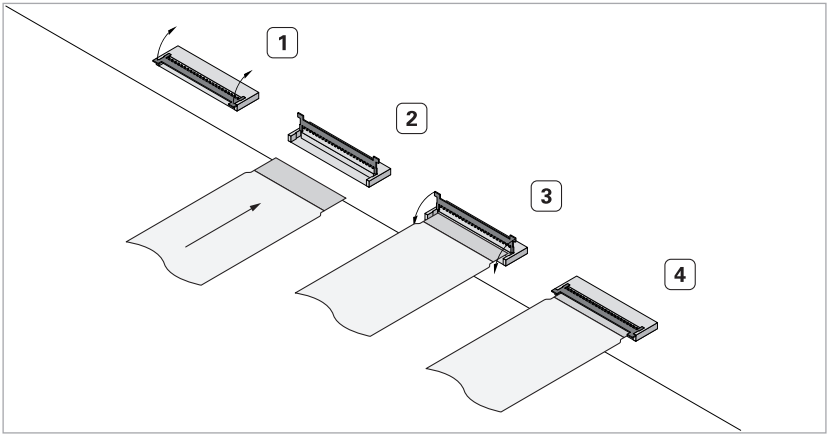


Fig. 11: Attaching the FPC cable to the FPC connector

5.3 Connecting the Interface Cables

The minimum requirements to access the camera are a functioning Ethernet connection and a power supply. Depending on your application, you might also need the serial connection, S-Video output, trigger/strobe and I/Os.

5.3.1 Standard Interface Board

Note

The following instructions only apply to the standard interface board and assume that you have ordered the optional cable and adapter kit "VRmDC Eval. Acc. Kit 2.0" (➲ section 3.4).

Connecting to Ethernet:

1. Connect an Ethernet cable (not supplied with the Eval kit) to the camera's Ethernet port (➲ Fig. 6 on page 13).
2. Connect the Ethernet cable to your local area network.

Note

Ethernet setup is described in sections ➲ chapter 6.1.

Connecting RS232, S-Video output, trigger/strobe and GPIOs

Note

Access to the camera's serial console is only possible using the separate DF14-15P interface on the CPU board and the service cable (➲ section 5.3.3). The RS232 interface of the MPE Garry Micro T connector described here can only be used to enable serial communication between the camera and external devices.

1. Connect the supplied interface adapter cable to the camera's Micro-T receptacle (➲ Fig. 6 on page 13). Secure the plug by fastening the screws attached to the plug. The interface adapter cable splits up into three separate cables (➲ Fig. 12 on page 26).
2. If you plan to connect an external device to the camera using RS232, connect the external device to the D-Sub plug coming from the adapter cable.
3. To view the camera's S-Video output, connect the supplied S-Video cable to the S-Video receptacle of the interface adapter cable. Connect the other end of the S-Video cable to your output device.
4. The remaining cable again splits up into 15 single receptacles for 0.1" connectors (trigger, strobe, composite, and general purpose I/Os).
   ▶ View the cable plan for a detailed description (➲ section 9.1.1)

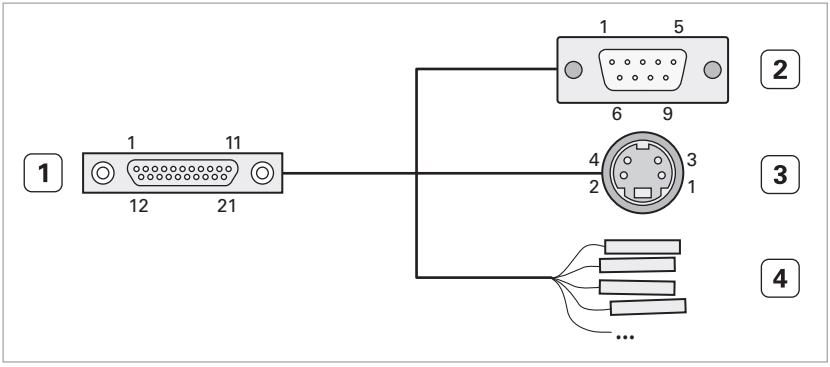


Fig. 12: Interface adapter cable

1 MPE Garry Micro T 387-1-021-ZS0, plugged into camera receptacle
2 D-Sub-9, male
3 S-Video, female
4 Receptacles for 0.1" connector (15 pieces)

5.3.2 CUEO1 Pico Interface Board

Note

The following instructions apply to the CUEO1 pico interface board only and assume that you have ordered the optional cable and adapter kit "VRmDC Eval. Acc. Kit PICO" or "VRmDC Eval. Acc. Kit PICO2" (➲ section 3.4).

Refer to ➲ section 9.1.3 for a pinout of the pico interface board.

If your camera is equipped with the CUEO1 pico interface board, all interfaces are available via miniature connectors (➲ Fig. 7 on page 14). Depending on the ordered Eval kit, the cables contained in the kits are equipped with a matching miniature connector on one end and open contacts or corresponding ports on the other end.

Connecting to the miniature interfaces with VRmDC Eval. Acc. Kit PICO:

1. Choose the matching cable for the interface you wish to connect to. For example, to connect to the 10-pin general purpose I/O interface labelled GPIO, choose a cable with a 10-pin connector.
2. Connect the cable to the corresponding connector on the camera's interface board.
3. Connect the other end of the cable to the target interface of your system.

Connecting to the miniature interfaces with VRmDC Eval. Acc. Kit PICO2:

1. Connect the 10-pin connector to the panel with the RJ45 port. Connect the 8-pin connector to the cameras's 8-pin Ethernet interface labelled ETH.
2. Connect the USB-A cable to the cameras's 4-pin USB interface labelled USB.
3. Connect the power cable to the cameras's 6-pin power interface labelled PWR.
4. For the remaining interfaces, proceed as described in the previous step-by-step instruction.

5.3.3 Connecting Serial Console and JTAG

The connection of serial console and JTAG is only necessary for debugging and update purposes. You may skip this step for now.

Connecting serial console and JTAG:

Note

The DF14-15P connector for serial console and JTAG is accessible on board level cameras only.

OEM/COB cameras with an FPGA module (VRmDFC-X, multi-sensor cameras) are equipped with a second DF14 connector located on the FPGA board (➲ Fig. 3 on page 10). This second connector is reserved for service purposes only.

1. Connect the supplied service cable to the lateral Hirose DF14-15P connector on the CPU board (➲ Fig. 6 on page 13).
2. Connect the supplied null modem cable to the D-Sub receptacle of the service cable.
3. Connect the null modem cable to your computer. If the computer has no serial port, use the USB-to-Serial adapter included in the VRmagic Eval Kit (Windows: installation of the driver necessary, ➲ section 4.4).
4. The JTAG connectors are located on the 2x7-pin header.
   ▶ View the cable plan for a detailed description (➲ section 9.1.2)

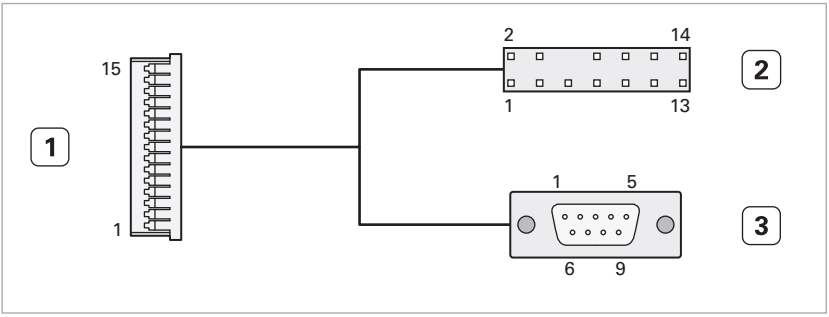


Fig. 13: Service cable for intelligent cameras

1 Hirose DF14-15S-1.25C, plugged into camera receptacle

2 2x7 header, 2.54 mm, key on pin 6

3 D-Sub-9, male

5.4 Connecting the Power Supply

Use the AC power adapter PHIHONG PSA 15R-050P included in the accessory kit. Alternatively, use a power supply with +5 V DC (±5%) and at least 15 W.

Connecting the camera to power:

1. Connect the power supply adapter to the AC mains.
2. Connect the power supply cable to the camera's power inlet (➲ section 3.3).
   ▶ The camera starts to boot up.
3. Wait for the status LED to light up.
   ▶ The camera is ready to use.

6 First Steps

Note

Some of the Linux commands in this manual require root user rights. If you are not logged in as root, you can type `sudo <command>` . The prefix `sudo` allows you to execute a single command with root rights after entering you Linux user password.

6.1 Setting Up the Ethernet Connection

Network with DHCP Server

The camera is configured to obtain an IP address from a DHCP server. If you have connected the camera to a network with a DHCP server, the Ethernet connection should be working. Proceed with ➲ section 6.1.2.

Network without DHCP Server

If no DHCP server can be found, the camera will fall back to an IPv4LL address. In this case, connect the camera directly to your computer's network interface (point-to-point connection). This requires the host computer to also fall back to an IPv4LL address (169.254.x.x). To achieve this, proceed with ➲ section 6.1.1.

Note

It is always possible to log in to your camera via RS232 (➲ section 6.3). This way you can identify and fix problems related to your specific network infrastructure.

6.1.1 Setting Up a Point-to-Point Connection

Note

This section only applies if your network is not equipped with a DHCP server.

Setting up a point-to-point Ethernet connection (Windows):

1. Configure the network interface to which the camera is connected so that an IP address and a DNS server address are obtained automatically.
   - ▶ This will fail after approximately one minute. The interface will fall back to an IPv4LL address on its own.

Setting up a point-to-point Ethernet connection (Linux):

1. Execute `avahi-autoipd -D eth[n]` on your host PC to obtain an appropriate IP address in the 169.254.x.x range.
2. Check the current IP address with `ifconfig`.

6.1.2 Enabling Ethernet Support and Checking the Connection

The Ethernet support for VRmagic devices is disabled by default (in CamLab as well as in the library). Follow these instructions to enable Ethernet support and check the connection by running the CamLab application, which is part of the VRmagic host SDK.

Enabling Ethernet support and checking the connection:

1. Enable Ethernet support:

   Windows: choose **Start** > **All programs** > **VRmagic** > **VRmagic Ethernet Support** > **Enable Plug & Play** to enable Ethernet support.

   Linux: execute `./usr/local/bin/vrm_ethernet_enable.sh` to enable Ethernet support.

2. Start the CamLab application (Windows: **Start** > **All programs** > **VRmagic** > **CamLab**).
   - ▶ Provided that the camera's status LED is on, your camera should be listed in the **device** list (model name and serial number).

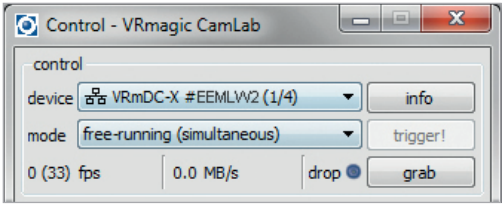


Fig. 14: Camera detected in CamLab

3. If not already the case, select your camera from the list.
   ▶ Your camera is connected to the Ethernet and ready to use.
4. If you wish to view the camera image, click **Grab**.
5. Leave CamLab open for the following steps.

6.2 Accessing the Device via SSH

If you have found the camera in CamLab, you should be able to access the camera using an SSH connection. To do this, you can use the Linux Terminal application. However, we recommend using PuTTY, which is an open-source SSH, Telnet, and serial client for Windows and Linux.

Finding out the camera's IP address:

1. To view the device's IP address, click the **Info** button in CamLab (➲ Fig. 14 on page 32).
   ▶ The device information window with the camera's IP address is displayed.

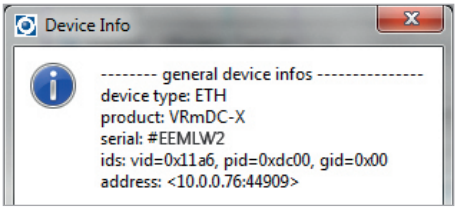


Fig. 15: Device Info window

Accessing the camera via SSH using PuTTY:

1. Start PuTTY on your host PC (➲ section 4.5).
2. In the **Session** category, choose **SSH** as **Connection type**.
3. Type in the camera's IP address and choose **Open**.
   ▶ The console window is displayed and prompts `login as:` .
4. Login as `root` and enter `vrmagic` as password. If you changed your password, enter your new password.
   ▶ You are now connected to the camera via SSH.

10.0.0.76 - PuTTY
login as: root
root@10.0.0.76's password:
Linux VRmagic-000-EEMLW2 2.6.27-davinci-VRm-1.52 #472 PREEMPT Wed May 16 09:
06:15 CEST 2012 armv5tejl
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
VRmagic-000-EEMLW2:~#

Fig. 16: SSH connection using Putty console

Accessing the camera via SSH using the Linux Terminal:

1. Open a terminal window on your host PC.
2. Execute `ssh root@<IP address>` .
   ▶ The console prompts `root@<IP address>'s password:`
3. Enter `vrmagic` as password. If you changed your password, enter your new password.
   ▶ You are now connected to the camera via SSH.

Note

For a short Linux command reference, view section ➲ section 9.2.

6.3 Accessing the Device via RS232

For most applications it is sufficient to connect via SSH. The serial console is only recommended for debug purposes and for updates using the VRmagic update flash drive.

Note

The serial console connector is accessible on board level cameras only. It is not accessible on PRO cameras in aluminium housing.

Requirements: The camera is connected to the host computer using the lateral DF14-15P connector (➲ section 5.3) and the service cable. A serial terminal application must be installed on the computer (for example minicom on Linux or putty on Windows or Linux).

Accessing the device via RS232:

1. Find out which serial port the camera is connected to (*Windows*: device manager, *Linux*: dmesg command). A typical value for Windows would be “COM1”, for Linux “ttyS0”.
2. Start the serial terminal software.
3. Establish a serial connection using the following settings:
   - 115200 Baud
   - 8 data bits
   - 1 stop bit
   - no parity
   - no flow control

   Windows/Linux (using Putty): In the **Session** category, choose **Serial** and specify COM port and speed. Configure the connection settings in the **Connection** > **Serial** category. Choose **Open** to enter the console and then press the Enter key once.
4. Login as `root` and enter `vrmagic` as password.
   ▶ You are now connected to the camera’s serial console.

6.4 Changing the Camera's Default Password

The user name for accessing the camera is "root" and the default password is "vrmagic". We strongly recommend changing the camera's default password.

Changing the camera's password:

1. Access the camera using SSH (➲ section 6.2).
2. Change the password by typing the command `passwd` and pressing <Enter>.
   ▶ You are prompted to specify the new password.
3. Enter the new passwords and press <Enter>.
4. Repeat the new passwords and press <Enter>.
   ▶ Your password has been changed.

6.5 Changing the Camera's IP Address

After booting the camera for the first time using a dynamic IP address (DHCP), you can switch to a static IP address by changing the camera's network configuration.

Note

If you do not assign a static IP address to the camera, the DHCP server may assign a different IP address to the camera every time the camera is switched on and off again. You can always find out the IP address with Camlab (➲ section 6.2).

Changing the network configuration:

1. Access the camera using SSH (➲ section 6.2).
2. Navigate to the **/etc/network/** directory by typing `cd /etc/network/` .
3. Open the **interfaces** file with the Nano text editor by typing `nano interfaces` .
4. Edit the file and change the configuration as desired. The Ethernet configuration can also be completely disabled by commenting out "auto eth0". For further information please refer to https://wiki.debian.org/NetworkConfiguration.
5. Save the file with <Ctrl>+<S>.
6. Exit the editor with <Ctrl>+<X>.

6.6 Exchanging Files and Data

For file exchange between the camera and a host computer, NFS or Samba/Windows shares can be mounted on the camera. Alternatively, you may exchange data using a USB flash drive.

Mounting NFS or Samba/Windows shares on the camera:

1. Create a network share on your host computer and grant access rights to a specific user (user name and password). You will need the user data in step 4.
2. Access the camera using SSH (➲ section 6.2).
3. Create a mount directory on the camera where the network share will be mounted to. Use the `mkdir` command to create a new directory. For example, to create a new directory "my_mount_dir" within the current directory, type
   `mkdir my_mount_dir`
4. Type in the following to mount the share:

   NFS:
   `mount <ip-address>:/<name-of-share> /<local-mnt-dir>`

   Samba/Windows:
   `mount //<name-of-host or ip-address>/<name-of-share> /<local-mnt-dir> -o username=<name>`

   ▶ You are prompted to enter username and/or password.
5. Type in the login data.
   ▶ The share has been mounted. You can view the contents by typing
   `ls /<local-mnt-dir>`

Explanations and Examples

`<ip-address>`	IP address of host computer containing network share
`<name-of-host>`	Name of host computer containing network share
`<local-mnt-dir>`	Directory on the camera where you wish to mount the share
`<name>`	User name needed to access the share

If you have created a network share named "share99" on a Windows host named "host99" and granted access rights to the user "user99". This share shall be mounted to the directory "mount99" on the camera. To mount this share, type
`mount -t cifs //host99/share99 /mount99 -o username=user99`

Mounting a USB flash drive:

1. Plug the USB flash drive into the camera's USB host port.
2. Access the camera using SSH.
3. Change to the **/dev** directory by typing `cd /dev` .
4. List the directory contents by typing `ls -l` .
   - The USB flash drive is probably listed as `sda`.
   - Each partition on the USB flash drive should be listed as `sda<X>` (X = partition number).
5. Create a local mount directory with mkdir, for example `mkdir /mnt/usb`
6. Now mount the USB flash drive to the local mount directory by typing `mount /dev/sda<X> /mnt/usb` (X = partition number)
   - You can now access the USB flash drive in the local mount directory.

6.7 Changing DSP Memory Size

Caution

Possible malfunction of DSP applications

Changing the size of the DSP memory means that you will have to adapt the memory maps of all DSP servers you would like to use on this camera. If you change the size of the DSP memory partition without adapting the memory maps, DSP (demo) applications will not work anymore.

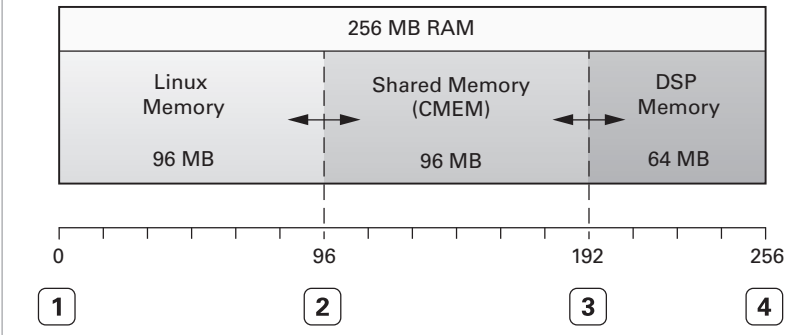


Fig. 17: Memory partitions (factory default values)

1 Start of main memory = Linux memory base = 0x8000 0000 (hex)

2 Start of shared memory (minimum at 60 MB,default at 96 MB)

3 Start of DSP memory (default at 192 MB, maximum at 256 MB); CMEM_END = this value - 1

4 End of main memory = 0x8FFF FFFF (for DR2) or 0x87FF FFFF (for DR1)

If you do not use the DSP, you can allocate this memory to the Linux and/or shared memory sections. This can be done by editing the CMEM_END variable in the **module_init_tools** file.

For example, to set the DSP memory size to 128 MB (0x0800 0000), set CMEM_END to (end of main memory - DSP memory size) = (0x8FFF FFFF - 0x0800 0000) = 0x87FF FFFF.

To change the Linux memory size, refer to ➲ section 6.8.1.

Changing the size of the DSP partition:

1. Access the camera using SSH (➲ section 6.2).
2. Navigate to the **/etc/init.d/** directory by typing `cd etc/init.d/` .
3. Open the **module-init-tools** file with the Nano text editor by typing `nano module-init-tools` .
   ▶ The CMEM_END variable occurs twice in the file.
4. Modify the hexadecimal CMEM_END variables to yield the desired DSP memory size and save the file.

6.8 Changing Kernel Boot Parameters

The Linux kernel gets its boot parameters from the bootloader (U-Boot). All of the following boot parameters can be altered using the **set_kernel_cmdline** script.

- Location of the root file system (NAND or NFS). Default: the camera boots from its internal NAND flash. NFS means that the camera boots from a network share.
- Video output mode (PAL, NTSC, LCD, or HDMI) and resolution.
- Console output device (Serial or Video). Default: Serial.
- Size of the Linux memory and shared memory (CMEM) partitions.

Caution

Possible malfunction of camera

False settings could prevent the camera from booting. Only change the boot parameters if you know what you are doing.

If something goes wrong, you have to connect to the camera's serial console (➲ section 6.3) and undo your changes directly in U-Boot by entering `nand resetenv` .

6.8.1 Partitioning Linux and Shared Memory (CMEM)

By setting the start of the shared memory (➲ Fig. 17 on page 38) you can modify the size of the Linux memory and at the same time of the shared memory (CMEM). The default size of the Linux memory is 96 MB.

Caution

Camera cannot capture images if no shared memory is left!

If you set the Linux memory to its maximum value, which is the start of the DSP memory (➲ Fig. 17 on page 38), then there is no memory left for the CMEM. Because images from the image sensor are always written to the CMEM section, this would result in the camera not being able to capture any images.

We strongly recommend not to change the memory partition settings if not really necessary. If you do, however, leave sufficient CMEM for image acquisition.

Changing the kernel boot parameters to change Linux memory size:

1. Access the camera using SSH (➲ section 6.2).
2. Type in `set_kernel_cmdline` and press Enter.
   ▶ The script starts. You are asked if you know what this is all about.
3. Choose "yes" by typing `1` and pressing Enter.
   ▶ You are asked to set the bootmode.
4. Choose "NAND" to let the camera boot from its local NAND memory (default).
   ▶ You are asked to configure the video output.
5. Choose a video output.
   ▶ You are asked to select a resolution.
6. Choose the desired resolution.
   ▶ You are asked to configure the console output device.
7. Choose "Serial" to output the console on the serial interface (default). Choose "Video" to output the console via the video output.
   ▶ You are asked if you would like to set the Linux memory size.
8. Choose "yes".
   ▶ You are prompted to enter the memory size.
9. Enter the Linux memory size in MB (➲ Fig. 17 on page 38). Keep in mind that changing the Linux memory size also changes the shared memory size.
10. Choose "OK" to apply the changes.

11. Reboot the camera by typing in `reboot` .
    - The camera now boots with the new boot parameters.

6.8.2 Changing Video Output Mode

The most common reason for changing the boot parameters is the selection of the video output mode, for example for connection of the optional HDMI output board to cameras with a pico interface board (VRmCUEO1).

Changing the video output mode for connection of an HDMI board:

1. Access the camera using SSH (➲ section 6.2).
2. Type in `set_kernel_cmdline` and press Enter.
    - The script starts. You are asked if you know what this is all about.
3. Choose "yes" by typing 1 and pressing Enter.
    - You are asked to set the bootmode.
4. Choose "NAND" to let the camera boot from its local NAND memory (default).
    - You are asked to configure the video output.
5. To make use of the external HDMI board, choose "HDMI".
    - You are asked to select a resolution.
6. Choose the desired resolution.
    - You are asked to configure the console output device.
7. Choose "Serial" to send the console output to the serial interface (default).
   Choose "Video" to output the console via the video output.
    - You are asked if you would like to set the Linux memory size.
8. Choose "no".
    - A summary of the configuration that will be applied after rebooting the camera is displayed.
    - You are prompted to proceed or to abort the whole procedure.
9. Choose "OK" to apply the changes.
10. Reboot the camera by typing in `reboot` .
    - The camera now boots with the new boot parameters.

6.9 Camserver and CamLab

A process called **vrmcamserver** is running on the camera. It is used by the VRmagic CamLab application to allow the transmission of images and to control the camera over Ethernet. By default, the camera can be accessed from anybody running the CamLab application or a custom vrmusbcam application in the same local area network.

Note

Camserver is a development tool. We recommend deactivating camserver once the camera is in a production environment.

If something does not work as expected, it is always a good idea to read the log file (**/tmp/vrmcamserver.log**).

The following steps require an SSH connection to the camera (➲ section 6.2).

Restricting access to the camera:

1. If you would like to allow access to the camera from specific hosts, change to the **/etc/vrmagic** directory on the camera's file system.
2. Edit the vrmcamserver.ini file by typing `nano vrmcamserver.ini` . View the comments in the file for further instructions.

Disabling the camserver process:

1. If you do not need the camserver, you can disable automatic start at boot time by entering `update-rc.d -f vrmcamserver.sh remove` on the camera's command line. Enter `update-rc.d vrmcamserver.sh defaults` to revert your changes and enable automatic start again.
2. If you merely want to disable the server temporarily (i.e. until the next boot cycle), enter `invoke-rc.d vrmcamserver.sh stop` on the command line.
   Enter `invoke-rc.d vrmcamserver.sh start` to start the server again.

6.10 Accessing the Camera's GPIOs

You can access the GPIO pins using the gpio-sysfs interface from the Linux command line. Before you can access a GPIO, you must export it into the user space.

Note

You can only access and control the GPIOs described in ➲ section 9.1.

Accessing the camera's GPIOs:

1. Access the camera using SSH (➲ section 6.2).
2. Export the GPIO you wish to access into the userspace by typing
   `echo <gpio_no> > /sys/class/gpio/export`

   Example for output 41: `echo 41 > /sys/class/gpio/export`
3. If you are accessing an output, you can set the output to high = 1 or low = 0.
   Set to 0: `echo low > /sys/class/gpio/gpio<gpio_no>/value`
   Set to 1: `echo high > /sys/class/gpio/gpio<gpio_no>/value`

   Example for output 41: `echo high > /sys/class/gpio/gpio41/value`
4. If you are accessing an input, first set the directrion to "in":
   `echo in > /sys/class/gpio/gpio<GPIO_NO>/direction`
   To read the state of the input, type
   `cat /sys/class/gpio/gpio<gpio_no>/value`

   Example for reading input 30: `cat /sys/class/gpio/gpio30/value`

6.11 Accessing the Camera's Temperature Sensor

The camera's internal temperature sensor is located directly next to the CPU. The sensor therefore does not measure the CPU temperature but the ambient temperature in the camera stack.

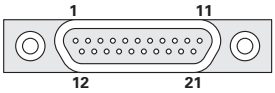
Accessing the camera's internal temperature sensor:

1. Access the camera using SSH (➲ section 6.2).
2. Type `cat /sys/bus/i2c/devices/1-0048/temp1_input` .
   ▶ If the output is, for example, 51250, then the temperature is 51.25°C.

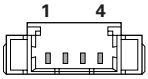
6.12 Using Trigger and Strobe

Depending on your interface board, the pins for trigger and strobe are located on different connectors. The following tables show the different connectors and the corresponding pinouts for trigger and strobe. If you are not sure about the type of interface board and the location of the connectors, please refer to ➲ section 3.3.

Cameras with Standard Interface Board and all Cameras with Housing

Connector	Pin	Signal
 MPE Garry Micro-T 386-2-021-ZS0	1	GND
	2	Passive Strobe Out – (3...24 V)
	3	Passive Strobe Out + (3...24 V)
	4	Active Strobe Out (+5 V)
	5	GND
	6	Passive Trigger In – (3...24 V)
	7	Passive Trigger In + (3...24 V)
	8	Active Trigger In (+3.3...5 V)
	9	GND
	10...11	+5 V
	12...21	...

Cameras with Pico Interface Board

Connector	Pin	Signal
 Molex 53398-0471	1	+5 V
	2	GND
	3	Active Trigger In (+3.3...5 V)
	4	Active Strobe Out (+5 V)

6.12.1 Trigger Input

Passive Trigger Input (potential-free)

The passive trigger input accepts a positive voltage (3...24 V). The signal will be interpreted as logically high when the voltage exceeds 3 V, and as logically low when the voltage drops below 1 V. The current flowing between Trigger In + and Trigger In – is approx. 2 mA at 3.3 V and approx. 5 mA at 24 V. The active and the passive trigger input are connected internally and cannot be controlled separately.

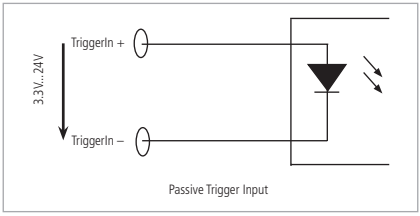


Fig. 18: Passive Trigger Input

Active Trigger Input (TTL)

An LVTTL/TTL compatible signal can be fed in at the active trigger input in reference to GND. The input features an internal pull-down resistor, input current max. 1 mA.

6.12.2 Strobe Output

Passive Strobe Output (potential-free)

The camera switches an external current between Strobe Out + and Strobe Out – using a potential-free, opto-coupled output transistor. The voltage difference between Strobe Out + and Strobe Out – must be positive (3...24 V). The output transistor is capable of switching currents of up to 100 mA. The voltage drop across it will never fall below 1 V.

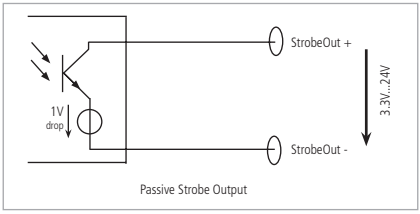


Fig. 19: Passive Strobe Output

Active Strobe Output (TTL)

At the active strobe out is a TTL compatible strobe signal referenced to GND. Active strobe out can sink or source currents of up to 20 mA.

6.13 Switching the Camera On and Off

We recommend the camera be switched off before disconnecting it from power. This allows the Linux OS to shut down properly.

Switching the camera on:

1. If the camera is connected to power but switched off, press the power button on the camera's interface board briefly to switch on the camera.
   ▶ The camera boots. When the status LED lights, the camera is ready.
1. If the camera is not connected to power, reconnect the camera to a power supply.
   ▶ The camera boots. When the status LED lights, the camera is ready.

Switching the camera off:

1. If you have access to the camera via SSH or RS232, type in `halt` to shut the camera down.
   ▶ After a few seconds, the camera shuts down. The status LED extinguishes.
2. A second option would be to use CamLab. Select the camera and press the **Shutdown** button on the **Misc** tab.
   ▶ After a few seconds, the camera shuts down. The status LED extinguishes.
3. Alternatively, press the power button on the camera's interface board briefly to shut down the camera.
   ▶ After a few seconds, the camera shuts down. The status LED extinguishes.

7 Programming Environment

Setting up the toolchains is necessary for compiling the demo applications and for writing your own applications. In both cases, the ARM toolchain must be installed. Installation of the DSP toolchain is optional.

Although the demos can be compiled by using nmake or gmake, we recommend that MS Visual Studio (at least Express) be used to work with them.

Best Practice for Directory Structure

Of course, you may install the toolchains and the VRmagic SDK where you wish. To avoid possible problems and to work more efficiently, we recommend installing the components in a directory structure that complies with these rules:

- Do not use special characters or space characters in path names. The use of the underscore character _ is unproblematic.
- For a better overview, install the components of each toolchain in common parent directories (➲ example below).
- Share the demo directory of the VRmagic SDK and the directory where you save your executables over the local area network. This way you can easily access your applications from the camera (➲ section 6.6).

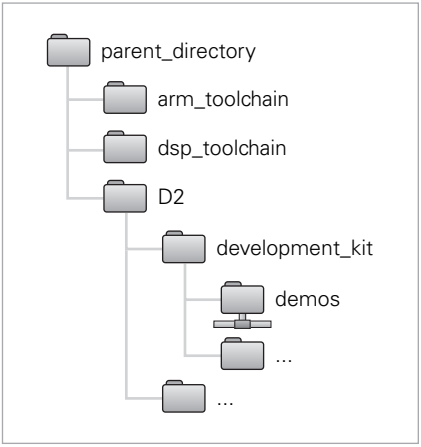


Fig. 20: Example directory structure on host PC

Note

The instructions provided in this chapter apply for both Windows and Linux, if not specified otherwise.

7.1 Setting Up the ARM Toolchain

Caution

During setup, choose an installation path without any space characters or hyphens. Space characters (e.g. “ARM toolchain”) will prevent the toolchain from working.

Windows: We recommend installing the toolchain in a separate directory and **not** in the Windows program directories (e.g. “program files”, “program files (x86)”)!

Installing the ARM toolchain:

1. Visit http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/
2. Scroll down to the **ARM Processors** heading and follow the link **Download the GNU/Linux Release**.
3. Follow the download procedure for **Sourcery CodeBench Lite Edition for ARM GNU/Linux**.
4. Once you are on the download page, download the version **2008q1-126**. for Linux or Windows.

Note

Full functionality of the demos has been tested with the release **2008q1-126**. If possible, download this release. Functionality with other versions is possible, but cannot be guaranteed.

5. Install the package to a directory of your choice. During setup choose **Modify PATH for all users**.
6. Extract the contents of **/D2/development_kit/toolchains/ARM/arm_toolchain_addons.zip**
   - You will receive a directory with several subdirectories and additional files for the Code Sourcery ARM EABI GNU/Linux toolchain, which is needed to compile code for the ARMv5T architecture (ARM9 processor) on x86 systems.

7. Add the contents of **arm_toolchain_addons.zip** to the respective directories in the toolchain's installation directory.
8. Restart your command shell if it is running.
9. To test the installation open a command shell (Windows: for Example cmd.exe) and execute
`arm-none-linux-gnueabi-gcc` .

   ▶ If the toolchain was set up correctly, the output should look like this and you can continue with ➲ section 7.2:
   `arm-none-linux-gnueabi-gcc: no input files`

   ▶ If you receive the message `command not found`, most probably the PATH environment variable was not set correctly in step 4. In this case, perform the following steps for your OS.

Setting the PATH environment variable (Linux):

1. Check the paths in your PATH environment variable. If necessary, add this path: **/your/path/to/Sourcery_G++_Lite/bin**
   ▶ Now you should be able to perform step 8 of the previous instructions.

Setting the PATH environment variable (Windows 7):

1. Choose **Start** > **Control Panel** > **System Properties** > **Advanced System Settings** > **Advanced** > **Environment Variables** .
2. In the **Environment Variables** window, double-click on the **Path** variable in the **System variables** list. You may have to scroll down.
   ▶ The **Edit System Variable** window is displayed (➲ Fig. 21 on page 53).
3. Check the paths in your PATH environment variable. If necessary, add this path: **\your\path\to\Sourcery_G++_Lite\bin**
   ▶ Now you should be able to perform step 8 of the previous instructions.

7.2 Setting Up the DSP Toolchain

Setting up the DSP toolchain is optional. However, in order to build the VRmagic DSP demos and your own C64x+ projects, you should complete the following steps.

Caution

During setup, choose an installation path without any space characters or hyphens. Space characters (e.g. “DSP toolchain”) will prevent the toolchain from working.

Windows: We recommend installing the toolchain in a separate directory and **not** in the Windows program directories (e.g. “program files”, “program files (x86)”)!

➲ See chapter 7 for further information.

Step 1: Creating a user account at TI.com for downloading software:

1. Visit http://www.ti.com/
2. Click on **my.TI Login** in the upper right corner.
   ▶ You should be redirected to a registration form.
3. Fill in and send off the registration form to create an account at ti.com.
   ▶ You are redirected to your profile page. You should have received an e-mail.
4. Follow the instructions in the e-mail.
   ▶ Your user account was created. You will need the user account to download the software in the following step.

Step 2: Downloading and installing packages:

1. Log in with your new my.TI account.
2. Download and install the following software. In some cases you might be asked to fill out further forms (U.S. Government export approvals).

Note

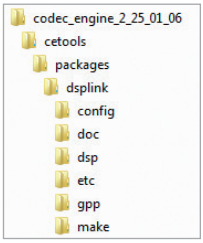
Full functionality of the demos has been tested with the software versions specified in the following instructions. Functionality with other versions cannot be guaranteed.

3. Download **Code Generation Tools v6.1.13** for *Windows* or *Linux* from https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm

4. Download **XDCtools 3.16.02.32** for *Windows* or *Linux* from http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/rtsc/index.html
5. Download **DSP/BIOS 5.41.03.17** for *Windows* (CCS4) or *Linux* from http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/bios/dspbios/index.html
6. Download **Codec_Engine 2.25.01.06 (with cetools)** from http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/ce/index.html
7. Optional: **JPEG Encoder, Version 2.00.01.00** for *Windows* or *Linux* (only for jpegenc_demo) from http://software-dl.ti.com/dsps/dsps_public_sw/codecs/C64XPlus_Video/index_FDS.html
8. Optional: **TI C64XPLUS-IMGLIB v2.02** (only for sobel_viewer demo) from http://software-dl.ti.com/dsps/dsps_public_sw/c6000/web/c64p_imglib/latest/index_FDS.html

Step 3: Installing VRmagic DSP toolchain add-ons:

1. Extract the contents of **/D2/development_kit/toolchains/DSP/dsplink.zip** .
   - ▶ You will receive a directory with several subdirectories and additional files for the Codec Engine.
2. Add the contents of **dsplink.zip** to the directory **/<your-path-to>/codec_engine_2_25_01_06/cetools/packages** .



Step 4: Replacing Linux Utils:

The Linux Utils, which are delivered by the Codec Engine 2.25.01.06 package have a bug in the ARM cache handling. Therefore the utils have to be replaced:

1. Delete the directory
   /<your-path-to>/codec_engine_2_25_01_06/cetools/packages/ti/sdo/linuxutils
2. Download **Linux Utils 2.25.02.08** from
   http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/linuxutils/index.html
3. Extract **linuxutils_2_25_02_08.tar.gz** to some temporary directory.
4. Copy the directory
   /<your-temp-path-to>/linuxutils_2_25_02_08/packages/ti/sdo/linuxutils
   to
   /<your-path-to>/codec_engine_2_25_01_06/cetools/packages/ti/sdo

Step 5: Adding PATH environment variables (Linux):

1. Add these paths to your PATH environment variable:
   /<your-path-to>/xdctools_3_16_02_32
   /<your-path-to>/xdctools_3_16_02_32/bin
   - Now you should be able to compile the demos from the **/D2/development_kit/demos/ARM_and_DSP** directory.

Step 5: Adding PATH environment variables (Windows 7):

1. Choose **Start** > **Control Panel** > **System Properties** > **Advanced System Settings** > **Advanced** > **Environment Variables** .
2. In the **Environment Variables** window, double-click on the **Path** variable in the **System variables** list. You may have to scroll down.
   - The **Edit System Variable** window is displayed.

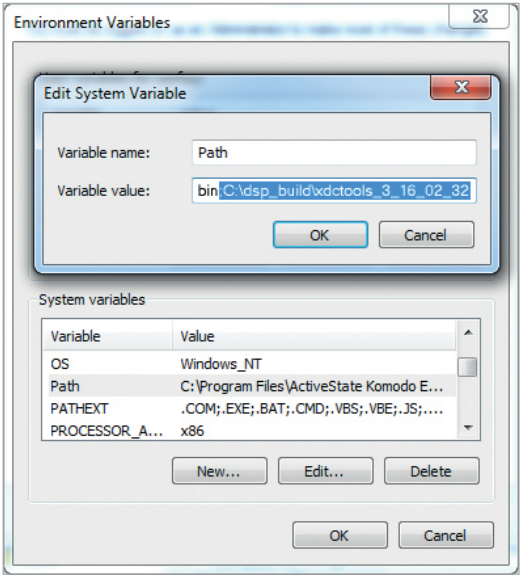


Fig. 21: Adding PATH variables in Windows 7

3. Now add the following paths, separating each path with semicolons:

\<your-path-to>\xdctools_3_16_02_32
\<your-path-to>\xdctools_3_16_02_32\bin

▶ Now you should be able to compile the demos from the **/D2/development_kit/demos/ARM_and_DSP** demos directory.

Note

Great sources of information about the DaVinci processor and the Codec Engine:

- http://wiki.davincidsp.com
- The free book "OMAP and DaVinci Software for Dummies" obtainable at http://www.ti.com/dsp/docs/dspsplash.tsp?contentId=52451

7.3 Update of an Older Development Kit

If you have downloaded a new development kit from the VRmagic website, follow these instructions to work with the new software.

Updating an older development kit:

1. Check the documentation of the new development kit (i.e. this document in a newer version).
2. Compare the toolchain components recommended in the new documentation with the ones you have installed. Have the release numbers changed? Are there any new components to be installed?
   If yes, remove the old tools and install the required release(s).
3. In any case, be sure to update the VRmagic add-ons for the toolchains:
   /D2/development_kit/toolchains/ARM/arm_toolchain_addons.zip
   (➲ section 7.1) and
   /D2/development_kit/toolchains/DSP/dsplink.zip
   (➲ section 7.2).
4. Install the new camera firmware as described in ➲ chapter 8.

7.4 Running Demo Applications

Copying the demos to a local directory:

1. We recommend sharing the **\D2\development_kit\demos** directory over the network, so that you can mount and access it on the camera's Linux OS as described in ➲ section 6.6.

7.4.1 ARM Demos

The ARM demos show how to use the VRmagic API. These demos run only on the ARM processor. You will find the ARM demos in **\D2\development_kit\demos\ARM**. The following demos are included:

- dc_gpio: Shows how to access the digital GPOs and GPIs of the camera. Refer to ➲ section 9.1 for pin assignments.
- deviceinfo: lists properties of all connected VRmagic devices.
- restorefactorydefaults: resets all settings of all connected VRmagic cameras to default.

- sdlviewer: displays the video images of the camera on the currently configured video output of the camera (S-Video, HDMI, LCD) using the SDL library.
- sdlviewer_cpp: the same as sdlviewer except that is has a C++ wrapper.
- vm_lib_demo: demonstrates the use of the VM_LIB image processing library for a simple object recognition.

Compiling the ARM demos on a Windows PC with MS Visual Studio:

The following steps assume that you have installed the ARM toolchain (➲ section 7.1) and MS Visual Studio (at least Express version).

1. In the file **Rules.mak** in the **demos/ARM** directory, adapt the path to the **D2** directory of the VRmagic D2 SDK.
2. Load the solution file **demosc++-davinci.sln** with Visual Studio and press F7 to build all demos.
3. To test the binaries on the camera, you can share the demo directory over the network and mount it on the camera (➲ section 6.6). Alternatively you can use an USB flash drive to exchange the files (➲ section 6.6).

Compiling the ARM demos on a Windows or Linux PC with gmake or cs-make:

1. In the file **Rules.mak** in the **demos/ARM** directory, adapt the path to the **D2** directory of the VRmagic D2 SDK.
2. With gmake (e.g. from xdctools installation) or cs-make (from ARM toolchain installation):
   To build all demos change to the ARM directory and type
   `[g]make CROSS_COMPILE=arm-none-linux-gnueabi-`
   - ▶ After the build process is finished, the executables can be found in the corresponding subdirectories.
   - ▶ If the build process does not work, please read the error messages displayed on the console.
3. To test the binaries on the camera, you can share the demo directory over the network and mount it on the camera (➲ section 6.6). Alternatively you can use an USB flash drive to exchange the files (➲ section 6.6).

Running the ARM demos:

1. Access the camera via SSH (➲ section 6.2).
2. To run an application, type
   `./path/to/executable/name_of_executable`

   If you already are in the same directory as the executable, simply type
   `./name_of_executable`

   Example: to run the deviceinfo application, type
   `./path/to/demo_directory/ARM/deviceinfo/deviceinfo`

7.4.2 ARM and DSP Demos

The ARM and DSP demos show how to use the VRmagic API and how to access the DSP on the intelligent camera. These demos use both the ARM and DSP processor. You will find the demos in **\D2\development_kit\demos\ARM_and_DSP**. The following demos are included:

- directfb_viewer: The DSP is used for image color conversion to RGB565. The ARM is used for framerate text overlay and displaying images on the currently configured video output (S-Video, HDMI, LCD).
- jpegenc_demo: This demo does the same as the directfb_viewer demo. In addition, the following action is performed everytime a key is pressed: the DSP converts the images to 8-bit gray format (or YUV in case of a color camera), performes JPEG encoding using the TI JPEG encoder, and saves the JPEG image to /tmp. Be aware that /tmp resides in the system RAM, so the images are deleted when the camera is switched off or rebooted.
- sobel_viewer: Uses the DSP to convert images to 8-bit gray format, applies the sobel filter from the TI DSP image processing library (TI C64XPLUS-IMGLIB), converts them to RGB565 and then uses the ARM to display the images on the currently configured video output of the camera (S-Video, HDMI, LCD).
- vm_lib_demo_davinci: Demonstrates the use of the VM_LIB image processing library on the DSP; images are converted to 8-bit-gray format and a blob detection is performed. The result is displayed on the currently configured video output of the camera (S-Video, HDMI, LCD).

Note

The demos have to be executed directly from the ARM_and_DSP directory. Otherwise, the DSP executable will not be found.

Compiling the ARM and DSP demos on a host PC:

The following steps assume that you have installed the ARM toolchain (➲ section 7.1) and the DSP toolchain (➲ section 7.2).

1. Adapt the paths to your toolchains and specify the intelligent camera board type ("DR1" or "DR2") in the following files:

   <**your-path-to>/demo_xyz/user.bld**
   Examples:
   ```
   c64p.rootDir= "d:/cam/dsp_build/TI_CGT_6_1_13";
   arm.rootDir= "d:/cam/arm_build/SourceryG++Lite";
   var VRmagicPlatform= "DR2";
   ```

   <**your-path-to>/demo_xyz/xdcpaths.mak**
   Examples:
   ```
   CE_INSTALL_DIR := d:/cam/dsp_build/codec_engine
   XDC_INSTALL_DIR := d:/cam/dsp_build/xdctools
   BIOS_INSTALL_DIR := d:/cam/dsp_build/bios
   VRMSDK_INSTALL_DIR := d:/cam/development_kit
   VRMAGIC_PLATFORM = DR2
   CGTOOLS_V5T = d:/cam/arm_build/SourceryG++Lite
   CC_V5T = bin/arm-none-linux-gnueabi-g++
   ```

Note

Cameras with board type VRmDR1 have 128 MB RAM, those with VRmDR2 have 256 MB RAM. To find out the board type, use the console command `dump_device_ini` . If your camera is equipped with the VRmDR2 board, you will read the following (amongst others):

```
[CPU_BOARD]
type = VRmDR2
```

2. Repeat step 1 for each demo.
3. IMPORTANT for sobel_viewer: Also adapt the path to your TI C64XPLUS-IMGLIB in the following files to compile the demo.

   <your-path-to>/sobel_viewer/server/link.cmd
   <your-path-to>/sobel_viewer/dsp/sobel/package.bld

   See ➲ section 7.2 if you have not installed TI C64XPLUS-IMGLIB yet.

4. IMPORTANT for jpegenc_demo: If you get the error `std.h: No such file or directory`, open the file

   <your-path-to>/JPEG-Enc/packages/ti/sdo/codecs/jpegenc/idmjpge.h

   and change `#include <std.h>` into `#include <xdc/std.h>`.
   See ➲ section 7.2 if you have not installed the JPEG Encoder yet.

5. Windows with MS Visual Studio (at least Express):
   Load the solution file ***.sln** in the respective demo directory with Visual Studio and press F7 to build the demo.
   - After the build process is finished, the executables can be found in the corresponding subdirectories.

6. Linux or Windows with gmake (e.g. from xdctools installation):
   To build a demo use the console and change to the **ARM_and_DSP/<demo>** directory.
   On Windows type `gmake -f nmakefile-davinci`.
   On Linux type `make`.
   - After the build process is finished, two executables can be found in the <demo> directory.
   - <demo>, which is the ARM executable and has to be executed on the camera.
   - <demo>.x64P, which is the DSP executable and has to reside in the same directory as <demo>. It is loaded and executed automatically on the DSP processor when the ARM executable is started.
   - If the build process does not work, please take a look at the error messages displayed on the console.

7. To test the binaries on the camera, you can share the demo directory over the network and mount it on the camera (➲ section 6.6). Alternatively you can use an USB flash drive to exchange the files (➲ section 6.6).

Running the ARM/DSP demos:

Note

The demos will not run if you are currently accessing the camera with CamLab.

1. Access the camera via SSH (➲ section 6.2).
2. To run an application, navigate to the directory containing the respective ARM and DSP executables by typing
   `cd path/to/executable`
3. After you have navigated to the directory containing the executables, type
   `./name_of_executable`
   to run the application.

8 Firmware Update and Recovery

8.1 General Description

You can update your camera's firmware if a newer version is available, or you can restore the original firmware image. The procedure is basically the same for both.

The firmware files can be found in the directory **\D2\firmware** of the VRmagic SDK. Currently there are two options to update your VRmagic camera.

Update via Ethernet (not suitable for recovery)

The update process via Ethernet is controlled by a host application, which runs on the host PC. The Ethernet update requires a camera with an intact firmware and a running SSH server (factory-default). It is therefore not suitable for recovery purposes.

Update/recovery via the USB update flash drive

This is method is the safest and also suitable for recovery. When the USB update flash drive is connected to the camera, the camera boots from the flash drive instead from the camera's NAND flash. After boot-up, it is possible to exchange all parts of the firmware.

Caution
Possible loss of data during update or recovery

Be aware that all data stored on the camera will be lost during the update process. To avoid data loss, you can either:

- Back up your data to another computer or storage device.
- Specify a list of files and directories that shall automatically be saved and re-stored after the update. This option is described in the following instructions.

8.2 Updating via Ethernet

Requirements

- The Ethernet updater uses the SSH protocol. Therefore it is required that an SSH server is running on the camera before the update is started. By default the Dropbear SSH server runs on every intelligent camera from VRmagic.
- The camera has to be connected to the network.

Performing updates via Ethernet:

1. Download the VRmagic SDK version you wish to install from http://www.vrmagic.com/en/imaging/downloads/ .
2. Unzip the file and change to **<your-path-to>/vrmagic-sdk/D2/firmware**.
3. Copy the following files to the **./ethernet_updater/firmware** directory:
   debian-root-ubifs.img
   ubl-vrmdr1.bin
   ubl-vrmdr2.bin
   u-boot-vrmdr1.bin
   u-boot-vrmdr2.bin
   uImage

Note

If you would like to use a custom kernel or root file system, you need to base these on the kernel/rootfs sources from the SDK version you would like to install.

SDK versions prior to 3.14 are not supported!

See **your-path-to/vrmagig-sdk/D2/firmware/src/[rootfs|kernel]/README.txt** for further instructions on how to build custom kernels or root file system images.

4. Edit the **./ethernet_updater/firmware/backup.txt** file to specify files and directories that should not be updated or overwritten during the update process. Each line in this file represents a file or a directory on the camera, which is backed up during the update process and automatically restored into the new root file system of the camera at the end of the update process. Be careful with this feature, since restoring, for example, an old VRmagic library to the new system could prevent the camera from working properly. Furthermore, you must be sure that the files and/or directories specified in backup.txt really exist, otherwise the backup will be cancelled and the update process will fail.

   Examples for entries in backup.txt:
   `/etc/vrmagic/cfg*` (saves settings stored on camera)
   `/your/path/to/directory/` (saves complete directory)
   `/your/path/to/file.xyz` (saves specific file)

5. Determine the IP address of the camera to be updated. There are three options for doing this:
   - Use the `ifconfig` command (SSH/Serial connection to camera necessary).
   - Use Camlab (➲ section 6.2)
   - Use the **./ethernet_updater/vrmdcethdiscover** command line utility. Linux users have to use the "mono" package to be able to use this tool.

   The latter two options require the vrmcamserver process to be running on the camera (➲ section 6.9), the VRmagic SDK to be installed (➲ section 4.3), and Ethernet support to be enabled (➲ section 6.1.2).

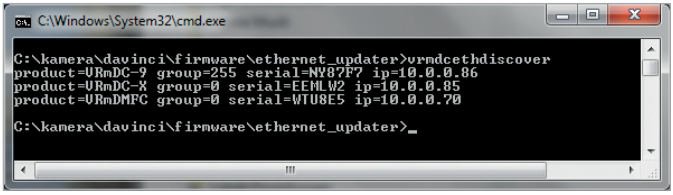


Fig. 22: vrmdcethdiscover tool

Caution

Possible damage to the device because of power failure.

If the device is disconnected from power during the update process, the camera firmware may become corrupted and the camera must be returned to VRmagic for service.

6. Execute the VRmDC-x Ethernet Updater on the Windows or Linux command line:

```
vrmdcethupdater.exe [options] <ip_of_camera>
```

One or more of the following options are supported:

`-f <path>`	relative path to firmware files (default: ./firmware)
`-p <pwd>`	SSH password of remote system (default: vrmagic)
`-q`	do not ask the user if he is sure about updateing
`-u <usr>`	SSH login name of remote system (default: root)

Note

There also is a batch file for Windows (./ethernet_updater/fw-update.bat), which automates the process described in steps 5 and 6.

- If a firewall is running on your host computer, most probably a window will pop up warning you about the update tool accessing the network. Allow access, otherwise the update process cannot be finished.
- If you did not use the -q option in step 6, a security question is displayed, which asks you if you are aware you will lose all data on the camera. Of course, the files and directories specified in step 4 will be restored after the update.

7. Answer with "yes" to continue with the update process.
   - The update process now runs automatically: the bootloader of the camera is prepared for the update. The camera is rebooted and loads a temporary Linux system from the host PC. After this system has been booted, all firmware files are loaded from the host computer.
   - Now the camera has all prerequisites needed and the update is actually started. Notice that this is a critical point in the update process: if something goes wrong before this point, the camera can be rebooted with the old firmware version without any problems. If a problem occurs afterwards (e.g. power failure), the camera will most likely not be bootable and has to be returned to VRmagic for service.
   - As soon as the update is finished, the camera boots with the new firmware version and will be ready for operation after a short time.
8. The environment of the bootloader is not deleted during the update process to ensure that the user settings are not lost (for example the video output settings). If you would like to reset the bootloader environment to its most current defaults, you can enter the following commands on the camera console:

```
flash_eraseall /dev/mtd0

reboot
```

Note

If something went wrong or if you simply want to check the update process, take a look at the **vrm-update-fw-<group_id>-<serial_no>.log** file in the **ethernet_updater/firmware** directory on your PC.

In addition, all messages of the vrmdcethupdater tool itself are logged into the file **vrmdcethupdater.log** .

8.3 Updating or Recovering via USB Update Flash Drive

A USB update flash drive allows to update the firmware of a D2 intelligent camera through the USB host port. A USB update flash drive is supplied with the optional VRmagic Eval Kit (➲ section 3.4). It is also possible to create your own USB update flash drive (➲ section 8.3.1).

There are two possible update methods.

- Manual update/recovery (➲ section 8.3.2): A serial connection via the camera's RS232 interface is required. This is only possible with board-level cameras (OEM/COB). The RS232 interface is not accessible on PRO cameras.
- Automatic update/recovery (➲ section 8.3.3): No serial connection is required, and it works on all cameras. This is the fastest method, however, the manual method permits more control over the update process.

Note

Recovery with original VRmagic USB update flash drive
The original VRmagic USB update flash drive supplied with the VRmagic Eval Kit comes with pre-installed firmware (see label on the flash drive). If you have not modified the flash drive contents, you may use it to recover the firmware specified on the flash drive label. In this case, skip the following section and continue with ➲ section 8.3.2 or ➲ section 8.3.3.

Recovery with own USB update flash drive
If you wish to use your own USB flash drive for recovery, create a USB update flash drive as described in the following section.

Update
If you wish to update the camera firmware to a newer version, create a USB update flash drive using the new firmware version as described in the following section.

8.3.1 Generating a USB Update/Recovery Flash Drive

Note

You can use any USB flash drive with a capacity of 2 GB or more, however, there can be differences between flash drives, and this may be the reason that one model works flawlessly while another does not. Therefore, it is a good idea to check a certain model before deploying the device.

Requirements

- A Linux PC or a Linux virtual machine is required. It is currently not possible to generate update flash drives using Microsoft Windows.
- USB flash drive or original VRmagic USB update flash drive.
- The following utilities must be installed on your system:
  - sfdisk
  - mkfs.vfat
  - mkfs.ext2
- You need root user rights on your system to execute the script, because the script uses commands like sfdisk and mount.
- You need to know the name of the Linux device file of the plugged-in USB flash drive (e.g. **/dev/sda**). The command **cat /proc/partitions** may be helpful to determine the name.

Caution

The following procedure will delete all data on the USB flash drive. If you are using the original VRmagic USB update flash drive, we recommend labelling it with the new firmware version you are about to install.

Generating a USB update flash drive:

1. Connect a USB flash drive to your Linux PC. The flash drive must not be mounted. If it is automatically mounted, unmount it, but leave it plugged in.
2. Download the VRmagic SDK version you wish to install on the camera from http://www.vrmagic.com/en/imaging/downloads/ .
3. Unzip the downloaded file.
4. To start the USB update flash drive generation, open a command shell.
5. Change to **<your-path-to>/vrmagic-sdk/D2/firmware/update_stick** .

6. Make sure you know the name of the Linux device file of the USB flash drive. Do not use a device file of a particular partition like **/dev/sda1**, instead use **/dev/sda**. Execute:

```
./make_update_stick <USB device file>
```

- After the script has finished, the USB flash drive is ready to use. Its usage is explained in the following two sections.
- The flash drive has two partitions, a FAT partition and an ext2 partition.

Partitions of the USB Update Flash Drive

The FAT partition contains the firmware files and the configuration file for the update script. This partition is accessible within Windows and Linux. To modify the behaviour of the update flash drive (e.g. enable automatic updating without user interaction) refer to the following two sections.

The ext2 partition contains the Linux kernel and the root file system. This partition is only accessible within Linux. It normally is not necessary to modify anything in this partition.

8.3.2 Manual Update

Because a serial connection via the camera's RS232 interface is required, manual updates are possible with board-level cameras (OEM/COB).

Perfoming manual updates using a USB update flash drive:

1. If you would like to specify the files and directories that shall not be updated or overwritten during the update process, edit the file **backup.txt** on the USB update flash drive. To do this, connect the flash drive to a Windows or Linux host.

   Each line in this file represents a file or a directory on the camera, which is backed up during the update process and automatically restored into the new root file system of the camera at the end of the update process. Be careful with this feature, since restoring, for example, an old VRmagic library to the new system could prevent the camera from working properly. Furthermore, you must be sure that the files and/or directories specified in backup.txt really exist, otherwise the backup will be cancelled and the update process will fail.

   Examples for entries in backup.txt:
   `/etc/vrmagic/cfg*` (saves settings stored on camera)
   `/your/path/to/directory/` (saves a complete directory)
   `/your/path/to/file.xyz` (saves a specific file)

2. Access the camera via the serial console (➲ section 6.3).

3. Connect the USB flash drive to the camera and (re)boot it.
   - The camera should now boot from the USB flash drive, which can be verified by checking that the login prompt reads "Update-USB". If this is the case, proceed to step 4.
   - If the USB update flash drive is not detected at boot time, the environment of the U-Boot bootloader is most probably mis-programmed. To restore it, press any key on the serial console after power-up to stop the automatic boot process. Then enter the following commands:

     `U-Boot > nand resetenv` (press <y> and <enter> to confirm)
     `U-Boot > reset`

     Now the USB flash drive should be detected correctly.

4. After bootup of Linux, login as `root`. The password is `vrmagic`.

5. Execute `vrm-update-fw.sh`.
   - The update procedure starts.
   - A security question is displayed, which asks you if you are aware you will lose all data on the camera. Of course, the files and directories specified in step 1 will be restored after the update.
6. Answer with "yes" to continue with the update process.
   - When the update is finished, the following message is displayed: `Update process successfully completed.`
   - If you experience any errors, try to exectute `vrm-update-fw.sh` once again.If this fails again, enter

     ```
     cat /tmp/vrm-update-fw.log
     cat /tmp/vrm-update-fw.klog
     ```

     and send the outputs to service@vrmagic.com
7. If the update was successful, shut down the camera using the `halt` command.
   - The camera shuts down. Wait until the status LED is switched off.
8. Disconnect the USB flash drive.
9. Switch the camera on again by pressing the power button.
   - The camera should now boot the new firmware. Wait for the status LED to light up. The camera is now ready.

8.3.3 Automatic Update

We recommend that this method be used if you need to update several VRmagic cameras. Also, be sure to update at least one camera with the manual method described in the previous section to ensure that the update process succeeds.

Performing automatic updates using the USB update flash drive:

1. Connect the USB update flash drive to your host computer. Windows should automatically recognize the flash drive as a removable disk. If you use Linux, mount the first partition of the flash drive (e.g. /dev/sda1) to a local directory.
2. Enable the automatic update by editing the configuration file **update.cfg** on the flash drive. Set the following:

   ```
   AUTOMATIC_UPDATE=1
   ```

3. If you would like to specify files and directories that should not be updated or overwritten during the update process, edit the file **backup.txt** on the flash drive.

   Each line in this file represents a file or a directory on the camera, which is backed up during the update process and automatically restored into the new root file system of the camera at the end of the update process. Be careful with this feature, since restoring, for example, an old VRmagic library to the new system could prevent the camera from working properly. Furthermore, you must be sure that the files and/or directories specified in backup.txt really exist, otherwise the backup will be cancelled and the update process will fail.

   Examples for entries in backup.txt:
   `/etc/vrmagic/cfg*` (saves settings stored on camera)
   `/your/path/to/directory/` (saves a complete directory)
   `/your/path/to/file.xyz` (saves a specific file)

4. Connect the USB flash drive to the camera and (re)boot it.
   - The camera should now boot from the USB flash drive.
   - The vrm-update-fw.sh script is executed automatically after the camera has booted.

5. Wait for the status LED on the backside of the camera to light up, indicating that the script was executed successfully.

6. Shut down the camera by pressing the power button briefly.
   - The camera shuts down. Wait until it is switched off.

7. Disconnect the USB flash drive.

8. Switch the camera on again by pressing the power button.
   - The camera should now boot the new firmware. Wait for the status LED to light up. The camera is now ready.

9. In the unlikely event that the red light does not light up after more than 5 minutes, the bootloader of the camera may be misconfigured. If possible, try to perform the manual update (➲ section 8.3.2).

Note

To avoid the accidental overwriting of your camera firmware, “disarm” your flash drive again by setting AUTOMATIC_UPDATE back to 0 in update.cfg .

9 Appendix

9.1 Cable Plans and Pinouts

9.1.1 VRmDC-X Interface Adapter Cable

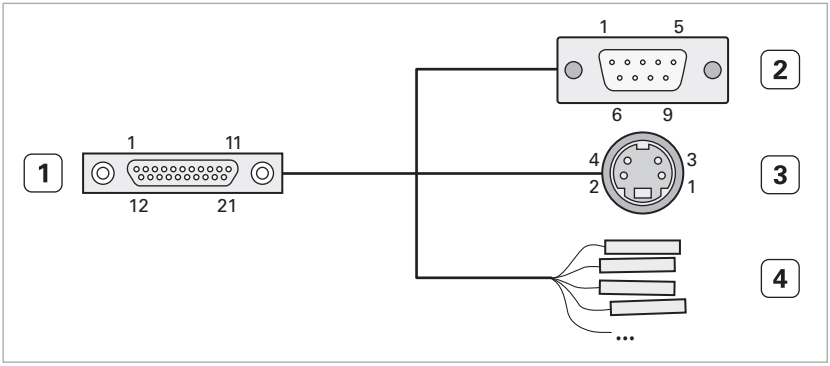


Fig. 23: Interface adapter cable for intelligent cameras

1 MPE Garry Micro-T 387-1-021-ZS0, to be plugged into camera receptacle

2 D-Sub-9, male

3 S-Video, female

4 Receptacles for 0.1“ connector (15 pieces)

#2 pin	#3 pin	#4 wire	#1 pin	Signal
2			18	User RS232 Receive
3			17	User RS232 Transmit
4			5	GND
	1 + 2		9	GND
	3		20	S-Video Y (DAC_B)
	4		19	S-Video C (DAC_C)
		➲ continued on next page		
		white	1	GND
		brown	2	Passive Strobe Out – (3...24 V)
		green	3	Passive Strobe Out + (3...24 V)

#2 pin	#3 pin	#4 wire	#1 pin	Signal
		yellow	4	TTL Active Strobe Out (+5 V)
		gray	6	Passive Trigger In – (3...24 V)
		pink	7	Passive Trigger In + (3...24 V)
		blue	8	TTL Active Trigger In (+3.3...5 V)
		red	10	+5 V
		black	11	+5 V
		purple	12	GP Output 41 [1]
		gray-pink	13	GP Output 37 [1]
		blue-red	14	GP Output 39 [1]
		white-green	15	GP Input 40 [1]
		green-brown	16	GP Input 30 (53 for 128MB) [1]
		yellow-white	21	Composite Video (DAC_A)

[1] +5 V TTL signal, max. 12 mA, 10K pulldown resistor

9.1.2 VRmDC-2 Service Cable

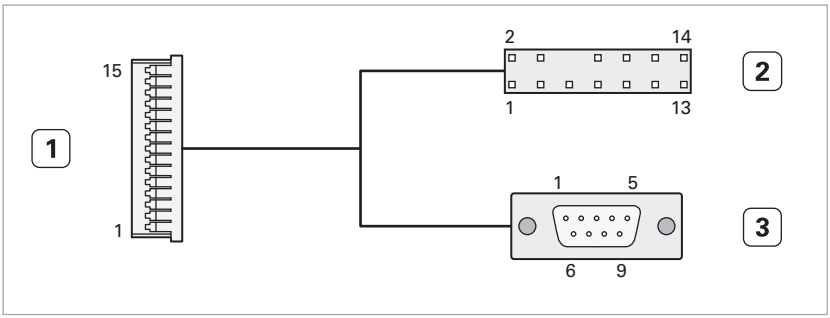


Fig. 24: Service cable for intelligent cameras

1 Hirose DF14-15S-1.25C, to be plugged into camera receptacle

2 2x7 header, 2.54 mm, key on pin 6

3 D-Sub-9, male

#2 pin	#3 pin	#1 pin	Signal
1		10	TMS (1.8 V level)
2		9	TRST# (1.8 V level)
3		8	TDI (1.8 V level)
4		7, 11	GND
5		6	TVD (1.8V)
6		-	(no pin)
7		5	TDO (1.8V level)
8		7, 11	GND
9		3	RTCK (1.8 V level)
10		7, 11	GND
11		4	TCK (1.8 V level)
12		7,11	GND
13		2	EMU0 (1.8 V level)
14		1	EMU1 (1.8 V level)
	2	15	Serial Console Receive
	3	14	Serial Console Transmit
	5	13	GND

9.1.3 VRmCUEO1 Pico Interface Board

The pico interface board is equipped with miniature connectors (Molex 53398-xx71). Each connector is identified by a label on the circuit board, for example „PWR“ for the power connection. Pin 1 of each connector is located next to the connector‘s label.

Note

The connectors are keyed. If the plug does not go in smoothly, be sure you have not inadvertently reversed the orientation.

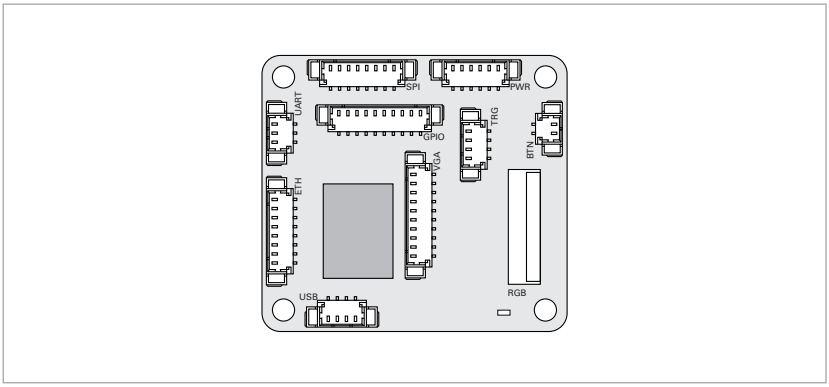


Fig. 25: Pico interface board

Power (PWR)	
Pin(s)	Signal
1-3	+5 V DC (±5%)
4-6	GND

Power button (BTN)	
Pin(s)	Signal
1	+5 V with onboard 1K2 series resistor
2	Power button

Ethernet (ETH)	
Pin(s)	Signal
1	+5 V
2	LINK LED [1]
3	ACTIVITY LED [1]
4	NC
5	RX-
6	RX+
7	TX-
8	TX+

USB host port (USB)	
Pin(s)	Signal
1	+5 V
2	D-
3	D+
4	GND

[1] Connect the LED between this pin and +5V (Pin 1). With onboard 470R series resistor.

SPI port (SPI)	
Pin(s)	Signal
1	+5 V
2	GND
3	GND
4	SPI_CLK/GPIO39 direction switchable by GPIO75 [2] [3]
5	SPI_EN0/GPIO37 direction switchable by GPIO76 [2] [3]
6	SPI_EN1/GPIO42 direction switchable by GPIO77 [2] [3]
7	SPI_DO/GPIO41 direction switchable by GPIO78 [2] [3]
8	SPI_DI/GPIO40 input only [3]

General Purpose I/O (GPIO)	
Pin(s)	Signal
1	+5 V
2	GND
3	GPIO79 (inout) [4]
4	GPIO80 (inout) [4]
5	GPIO81 (inout) [4]
6	GPIO82 (inout) [4]
7	GPIO83 (inout) [4]
8	GPIO84 (inout) [4]
9	GPIO85 (inout) [4]
10	GPIO86 (inout) [4]

[2] 0: ouput; 1: input (default)

[3] +5 V TTL signal, max. 32 mA (sink or source), 10K pulldown resistor

[4] +5 V TTL signal, max. 12 mA (sink or source), 10K pulldown resistor

Trigger/Strobe (TRG)	
Pin(s)	Signal
1	+5 V
2	GND
3	TTL Active Trigger in [5]
4	TTL Active Strobe out [6]

Serial port (UART)	
Pin(s)	Signal
1	TX [7]
2	RX [7]
3	GND
4	

[5] short-circuit this pin with +5 V (Pin 1) to trigger an image

[6] +5 V TTL signal, max. 32 mA (sink or source)

[7] +5 V TTL signal

Analog video output [8] (VGA)	
Pin(s)	Signal
1	I2C CLOCK
2	I2C DATA
3	GND
4	VSYNC
5	HSYNC
6	GND
7	Composite Video
8	S-Video Y
9	S-Video C
10	GND

[8] only PAL and NTSC timing are supported

9.2 Linux Short Reference and Examples

For a detailed Linux command reference we recommend downloading one of the readily available reference sheets available on the Internet. To view the manual page of any Linux command, type `man <COMMAND>` .

Hot keys

<Arrow up>	Repeat last command
<ctrl> + <r>	Search in recent commands

File comands

`cd example`	Change to subdirectory “example”
`cd example/dir`	Change to subdirectory “example/dir”
`cd`	Change to home directory
`cd /`	Change to root directory
`cd ..`	Change to parent directory
`ls` or `ls -l`	List directory contents (simple or detailed view)
`mkdir example`	Create new directory “example” in current directory

Viewing and Editing Files

`nano file.xyz`	View and/or edit the file “file.xyz” with the Nano editor

Network

`ifconfig`	List IP addresses for all devices on the local machine
`ping host`	Ping “host” and output results

Stopping and Starting

`halt -p`	Stop all processes and shut down with poweroff
`reboot`	Stop all processes and then reboot

User and Password

`passwd`	Change user password
`sudo <command>`	Execute single command with root user rights

10 Index

A

Access camera *32*
- RS232 *34*
- SSH *32*

Accessories *15*
ARM Toolchain *48*

B

Battery *7*

C

Cable plans *70*
Cable Plans *70*
Cables
- Interface Adapter Cable *70*

Camera architecture *8*
CamLab *31*, *42*
Camserver *42*
Connect to Ethernet *25*
Connect to power *29*
Connect to serial console *28*

D

Demo applications *54*
- ARM and DSP Demos *56*
- ARM Demos *54*

Directory structure *17*, *47*
Download software *16*
DSP Toolchain *50*

E

ESD prevention *7*
Ethernet Connection *30*
- Ethernet Support *31*
- Point-to-Point *31*
- Setup *30*

Eval kit *15*
External HDMI board *23*, *41*
External sensor boards *21*

F

File exchange *36*
Firmware recovery *60*
Firmware update *60*
- Ethernet update *61*
- USB Rescue Flash Drive *64*

FPGA board *10*, *12*

G

GPIOs *43*

H

HDMI output board *23*, *41*

I

Install console *20*
Install drivers *19*
Install SDK *18*
Interface adapter cable *15*, *26*, *70*
Interface board *10*, *11*, *12*
- Pico *14*, *27*, *73*
- Standard *13*, *25*

IP address *32*, *35*

J

JTAG *28*

K

Kernel boot parameters *39*

L

Linux commands *76*

M

Memory *9*
- CMEM size *40*
- DSP memory size *38*
- Linux memory size *40*
- Partitions *9*, *38*

Mount network share *36*
Mount USB flash drive *37*
Multi-sensor camera *12*

N

Null modem cable *15*

P

Password *35*
Pico interface board: *See* Interface board
Power *29*
Power button *13*, *14*, *46*
Power inlet *13*, *14*

R

Remote-sensor camera *11*
RGB888 interface *14*
RJ45 connector *13*
RS232 *34*

S

Safety *7*
Sensor board *10*
Sensor connection board *11*, *12*
Serial console *28*
Service cable *15*, *28*, *72*
set_kernel_cmdline *39*
Single-sensor camera *10*
SSH *32*
Standard interface board: *See* Interface board
Status LED *13*, *14*
Strobe *44*, *45*
Switch on/off *42*

T

Temperature Sensor *43*
Trigger *44*

U

Update: *See* Firmware update
USB host connector *13*, *14*
USB/serial adapter *15*
USB/serial adapter driver *19*
USB update flash drive *15*, *64*
- Automatic update *68*
- Generate new flash drive *65*
- Manual update *67*

VRmagic GmbH
Augustaanlage 32
68165 Mannheim
Germany
Phone +49 621 400 416-20
Fax +49 621 400 416-99

info.imaging@vrmagic.com
www.vrmagic-imaging.com

© 2014 VRmagic GmbH, Mannheim